NOAA TR NOS 60

A UNITED STATES
DEPARTMENT OF
COMMERCE
PUBLICATION

NOAA Technical Report NOS 60

U.S. DEPARTMENT OF COMMERCE
National Oceanic and Atmospheric Administration
National Ocean Survey

The Reduction of Photographic Plate Measurements for Satellite Triangulation

ANNA-MARY BUSH

ROCKVILLE, MD.
June 1973

NOAA TECHNICAL REPORTS

National Ocean Survey Series

The National Ocean Survey (NOS) provides charts and related information for the safe navigation of marine and air commerce. The survey also furnishes other earth science data--from geodetic, hydrographic, oceanographic, geomagnetic, seismologic, gravimetric, and astronomic surveys, observations, investigations, and measurements--to protect life and property and to meet the needs of engineering, scientific, defense, commercial, and industrial interests.

Because many of these reports deal with new practices and techniques, the views expressed are those of the authors and do not necessarily represent final survey policy. NOS series NOAA Technical Reports is a continuation of, and retains the consecutive numbering sequence of, the former series, Environmental Science Services Administration (ESSA) Technical Reports Coast and Geodetic Survey (C&GS), and the earlier series, C&GS Technical Bulletins.

Those publications marked by an asterisk are out of print. The others are available through the Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402. Price as indicated. Beginning with 39, microfiche is available at the National Technical Information Service (NTIS), U.S. Department of Commerce, Sills Bldg., 5285 Port Royal Road, Springfield, Va. 22151. Price $0.95. Order by accession number, when given, in parentheses.

COAST AND GEODETIC SURVEY TECHNICAL BULLETINS

*No. 22 Tidal Current Surveys by Photogrammetric Methods. Morton Keller, October 1963.

*No. 23 Aerotriangulation Strip Adjustment. M. Keller and G. C. Tewinkel, August 1964.

*No. 24 Satellite Triangulation in the Coast and Geodetic Survey. February 1965.

*No. 25 Aerotriangulation: Image Coordinate Refinement. M. Keller and G. C. Tewinkel, March 1965.

*No. 26 Instrumented Telemetering Deep Sea Buoys. H. W. Straub, J. M. Arthaber, A. L. Copeland, and D. T. Theodore, June 1965.

*No. 27 Survey of the Boundary Between Arizona and California. Lansing G. Simmons, August 1965.

*No. 28 Marine Geology of the Northeastern Gulf of Maine. R. J. Malloy and R. N. Harbison, February 1966.

*No. 29 Three-Photo Aerotriangulation. M. Keller and G. C. Tewinkel, February 1966.

*No. 30 Cable Length Determinations for Deep-Sea Oceanographic Operations. Robert C. Darling, June 1966.

*No. 31 The Automatic Standard Magnetic Observatory. L. R. Alldredge and I. Saldukas, June 1966.

ESSA TECHNICAL REPORTS

*C&GS 32 Space Resection in Photogrammetry. M. Keller and G. C. Tewinkel, September 1966.

*C&GS 33 The Tsunami of March 28, 1964, as Recorded at Tide Stations. M. G. Spaeth and S. C. Berkman, July 1967.

*C&GS 34 Aerotriangulation: Transformation of Surveying and Mapping Coordinate Systems. Melvin J. Umbach, August 1967.

*C&GS 35 Block Analytic Aerotriangulation. M. Keller and G. C. Tewinkel, November 1967.

*C&GS 36 Geodetic and Grid Angles--State Coordinate Systems. Lansing G. Simmons, January 1968.

*C&GS 37 Precise Echo Sounding in Deep Water. George A. Maul, January 1969.

*C&GS 38 Grid Values of Total Magnetic Intensity IGRF--1965. E. B. Fabiano and N. W. Peddie, April 1969.

C&GS 39 An Advantageous, Alternative Parameterization of Rotations for Analytical Photogrammetry. Allen J. Pope, September 1970. Price $0.30 (COM-71-00077)

C&GS 40 A Comparison of Methods for Computing Gravitational Potential Derivatives. L. J. Gulick, September 1970. Price $0.40 (COM-71-00185)

(Continued on inside back cover)

U.S. DEPARTMENT OF COMMERCE
Frederick B. Dent, Secretary

NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION
Robert M. White, Admiinistrator

NATIONAL OCEAN SURVEY
Allen L. Powell, Director

NOAA Technical Report NOS 60

The Reduction of Photographic Plate Measurements for Satellite Triangulation

ANNA-MARY BUSH

Geodetic Research and Development Laboratory
ROCKVILLE, MD.
JUNE 1973

For sale by the Superintendent of Documents, U.S. Government Printing Office,
Washington, D.C. 20402 - Price: $2.10, domestic postpaid; $1.75, GPO Bookstore
Stock No. 0317-00166

UDC 528.722.8:528.341:629.783:681.326

528	Geodesy
.341	Dynamic triangulation
.722	Equipment for reduction of photograms
.8	Comparison of plate coordinates
629.783	Navigation satellites
681.3	Data processing and equipment
.326	Computer programs

CONTENTS

THE REDUCTION OF PHOTOGRAPHIC PLATE MEASUREMENTS FOR SATELLITE TRIANGULATION

Anna-Mary Bush
Geodetic Research and Development Laboratory,
National Ocean Survey, NOAA, Rockville, Md.

ABSTRACT. The reduction of Satellite Triangulation raw (photographic) data and their preparation for analysis is described. The reduction method is closely related to the technique of plate measurement. The operations are carried out by means of three computer programs. The first sets up the framework for identifying images and computes a first approximation to camera orientation. The second program reduces the star and satellite image measurements to a single, consistent, plate-centered system, with comparator and operator biases removed. The third program associates each image with the celestial coordinates of the star and the time of exposure.

1. INTRODUCTION

1.1 Purpose of Program

The plate data reduction system is the first phase of data analysis for the Satellite Triangulation project of the National Geodetic Survey (Schmid 1972). Its purpose is to take the raw data which come in the form of a series of rectangular coordinates defining images on the photographic plate, and prepare them to be used as input to the Single Camera Orientation and Satellite Image Reduction programs. The Single Camera Orientation phase of satellite triangulation computes a mathematical model of a camera and its orientation in space, given the celestial coordinates of stars being photographed and the corresponding positions of star images on the photographic plate. The Satellite Image Reduction phase uses as input the camera parameters computed in the Single Camera Orientation program, and the positions of satellite images on the plate. (Slama 1972; Hanson 1973)

1.2 Historical Development of Program

The many operations necessary to reduce the raw data have evolved over a period of several years. During this time the results to date were analyzed repeatedly to detect biases, which were then eliminated or compensated for by changes in method. Originally, much of the reduction was done manually with the aid of several one-step computer programs written for the IBM 1620. As operations were added and the procedure became more complex, the number of such computer aids multiplied. The availability of the IBM 7030 made it possible to consolidate many of these previously separate steps, and turned the efforts of those in charge of data processing toward increasing automation. Automation also was made necessary by the great number of plates to be processed – about 3,000 altogether.

A total reevaluation of the plate data reduction procedure was made in 1968. At this time the procedure was cast in its final form. (All plates processed to date were rerun under the new procedure.) The various steps of the reduction operation were now consolidated into three computer programs written for the CDC 6600. A high degree of automation has been attained, but several steps of manual intervention and auxiliary computer output are still needed. Although the main emphasis of this report will be on the computer programs, the other steps will be described as needed.

The computer programs now used bear the traces of their evolutionary development. In many of the operations to be discussed, history rather than theory will be cited in the explanation. Also, the data formats now used have been influenced by the various input/output media used over the years.

1.3 Plate Measurement

The plate data reduction procedure is closely tied to the technique used to measure the photographic plate. This technique has been described by Schmid (1972). Certain aspects will be described here.

1.3.1 Exposure Times

A selected event consists of the simultaneous exposure of photographic plates at two or more observing stations. Each developed plate arrives from the field accompanied by a field record sheet, and a brush tape containing a record of the times at which the camera[†] shutter was open. These times are in a pattern. The pattern has been developed to obtain: i) clear star images before and after satellite passage to provide a firm basis for computing camera parameters and also to detect any camera motion during exposure; ii) star images during or as close as possible to satellite passage; iii) a series of evenly spaced satellite images passing across the plate close to plate center. The total exposure sequence is about 45 minutes. Because the camera is held fixed, each star in its field is represented on the plate by a series of images in the form of dots and dashes corresponding to the exposure pattern. The satellite, whose apparent speed is much greater than that of the stars, is seen as a line of evenly spaced dots. Regular breaks in the satellite exposure sequence are visible as gaps, and are used to associate each image with the appropriate time.

The scheme used for numbering images is of paramount importance, since this is the means by which each image is associated with time. For the stars, a basic code of five digits has been established (fig. 1). The leftmost two digits identify an individual star by location (see sec. 1.3.2 and sec. 1.3.3). The hundreds position indicates a particular part of the star's track length. Only the "dot" images in the star's track are used for measurement, and in the exposure pattern these are in nine consecutive groups The first three groups were exposed before satellite passage, the fourth as close before passage as possible, the fifth during passage, the sixth immediately after, and the final three after this. These groups have been given the (rather misleading) name of "trails." The tens and units positions together are in a running sequence beginning with the first exposure in trail 1. The usual case is to have five images in each trail.

It can be seen that the rightmost three digits of the star image number are time-related since they associate the image with its place in the exposure pattern. This part of the number will be called the time code. A given time code is represented somewhere in the track length of every star. The addition of the location code specifies the particular star.

The time code can also be used to refer to the star exposure times themselves. The times are transcribed from the brush tape and punched into cards along with their time codes. The exposure time of the initial satellite image is also recorded. All times are given as Greenwich Mean Time (GMT), also called Universal Time (UT).

[†] The main body of this report deals with the BC-4 camera. See appendix B for adaptations necessary when other cameras are used.

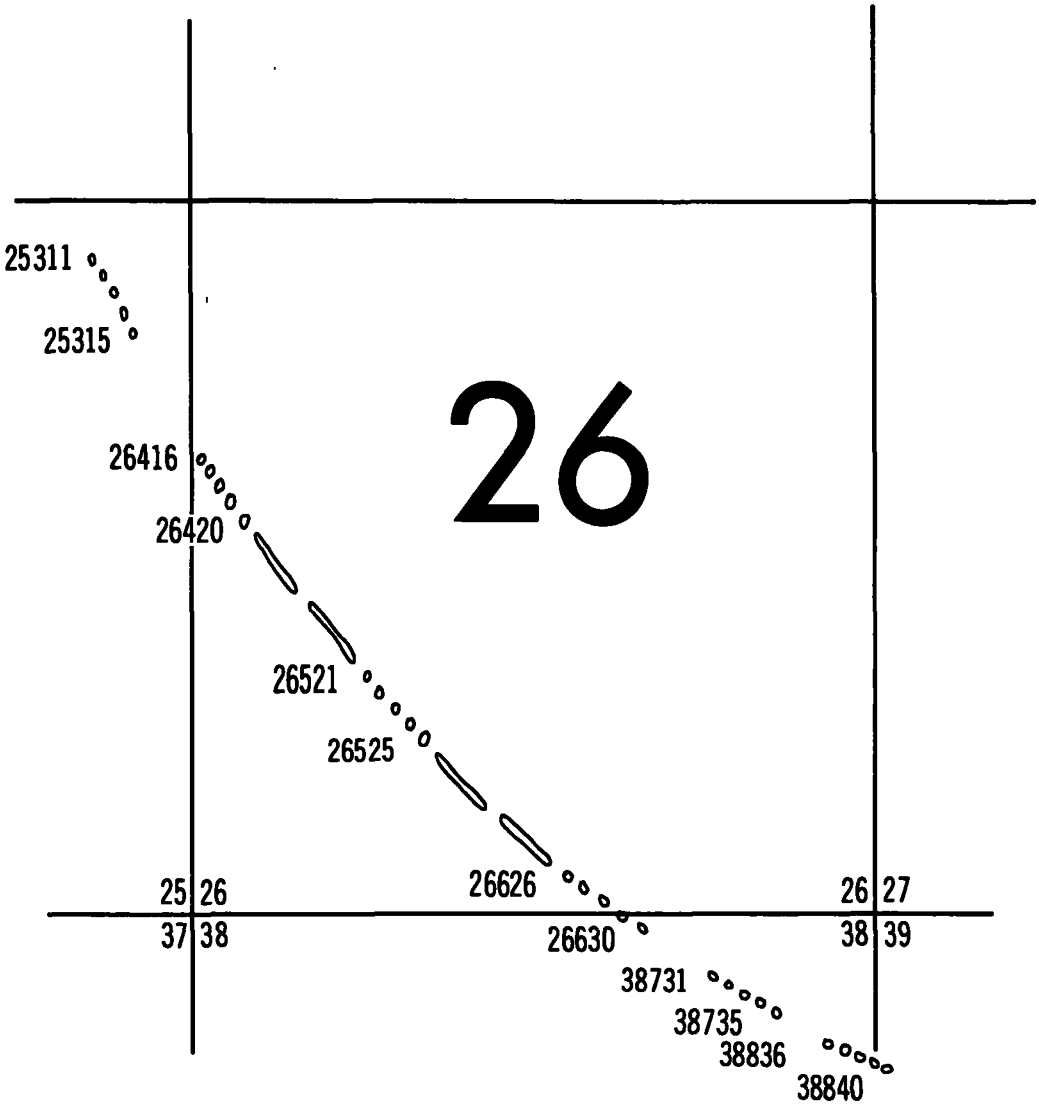


Figure 1. A section of a star's track with image numbers

1.3.2 Star Chart and Preidentified Stars

The other preparatory step done at this time is the automated preparation of a star chart to the scale of the photographic plate. This serves as a guide in measuring and identifying stars. At the same time, a group of bright stars in a ring around the center is selected to serve as preidentified stars. An arbitrary two-digit number that serves as location code is assigned to each of these. The celestial coordinates, plate coordinates, and location code of each are printed, enabling the operator to locate any of these stars on the plate. Also, the celestial coordinates and proper motions, together with the location code, are punched onto cards to be used later in the Preliminary Reduction program.

1.3.3 Marking and Measuring

The plate was preexposed with a 76-block grid. During the marking phase, about 150 stars are chosen for measurement, ideally, two from each grid block. Usually, only one trail from each star is to be measured, but if two or more trails from the same star are selected, this is allowed for in the reduction system.

The satellite images are related to the other plates in the event by the correspondence in brush tape times. The earliest image on any plate is assigned number 1, and all images, including missing images, are numbered in strict sequence from this. The operator selects five images, including the first and last images, along the satellite trail of the current plate, and marks them with their sequence numbers, which are readily established by their relation to the gaps in the trail. These are the preidentified satellites.

The drill holes in the photographic plate are numbered according to its orientation in the camera. Figure 2 shows the plate as it appears in the camera with the emulsion side away from the viewer.

The plate is placed in the comparator for measurement. As each image is centered in the eyepiece, the operator presses a button to record on paper tape the x and y coordinates to the nearest micrometer. With each position record the operator also adds the image class and a partial image number. The class is indicated by a code in which 3 means drill hole, 2 is star and 9 is satellite. See section 4.5.5 for discussion of why the operator does not number each image completely.

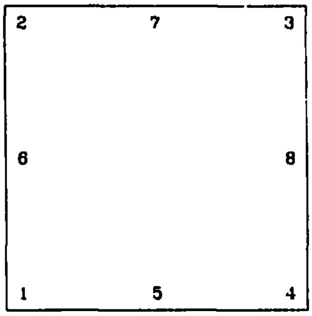


Figure 2. Drill holes

Study of measurements made under various conditions showed that after two to three hours the operator's body heat causes an unacceptable amount of error in the comparator. Therefore, the measurement of an entire plate, which takes up to eight hours, must be broken down into several sessions. At the beginning and end of each session the operator measures the eight drill holes, usually measuring each one five times before going on to the next.. At the end of the session the drill hole positions must be within a certain tolerance of those at the beginning; otherwise, the session's work must be repeated.

The first session is devoted to the measurement of the "Pre set" (see sec. 3). Besides the drill holes, the Pre set consists of the preidentified stars and satellites. The image number recorded for each satellite is the predetermined sequence number. The location code and trail number are recorded for each star, but the last two digits of the image number are given as zero.

During the next one or more sessions, all the star images previously selected and marked, and all the satellite images in the center 12 centimeters of the plate, are measured. This is called the A set. The image number recorded for each star is three digits only, the location code and trail number. The location code is the number of the block in which the star was measured. As in the Pre stars, all images in the measured trail of a given star have the same number at this point. No image numbers are recorded for the satellite images.

When the A set is complete, the plate is rotated 180° and the measurements are repeated to form the B set. Since each comparator operator has a personal bias in lining up a finite dot with a crosshair, repeating every measurement with the plate rotated 180° and taking a mean of the two sets will eliminate this bias.

The data on paper tape are converted to magnetic tape on the IBM 360. During the conversion, the drill hole measurements in the A and B sets are rearranged. For each measuring session, the opening and closing drill holes are combined; that is, all the measurements of drill hole 1 are grouped together, then all measurements of drill hole 2, and so on. These are placed ahead of the star and satellite measurements. These in turn are preceded by a set of class 0 images, originally representing fiducial marks. Later, the fiducials were dropped and the class 0 points are now a duplicate of certain drill holes (see sec. 4.3).

1.3.4 Independent Day Numbers

Besides the plate measurements, the other source of information is the list of exposure times from the brush tape. To compute the exact position of each star at each of these times (which are in UT), a set of parameters for each time, including Independent Day Numbers and local sidereal time, is necessary. Tables containing this information have been taken from the American Ephemeris and Nautical Almanac and stored on magnetic tape. With the station longitude and the date of observation given, a computer routine performs a lookup and interpolation for each exposure time, and writes the resulting set of parameters on tape. Each set of parameters includes:

time code, plate number, date
station latitude
local sidereal time of exposure

$$\left.\begin{array}{l} G\\ H\\ f\\ g\\ h\\ i\\ \tau \end{array}\right\}\quad \text{Independent Day Numbers}$$

ΔT = nearest Besselian year − 1950.0

$$\left.\begin{array}{l} \text{station elevation}\\ \text{atmospheric pressure and temperature} \end{array}\right\}\quad \text{from field record}$$

The computer routine also tabulates the exposure pattern, which is needed because the pattern may depart from the standard of nine trails with five images each. This is done by analyzing the time codes actually present. For each trail represented, the number of images and the sequence number of the first image are punched onto a card. This information is later used in numbering individual star images within a trail.

2. BASIC FORMULATIONS USED IN PLATE DATA REDUCTION

2.1 Some Considerations in the Reduction of Plate Measurement Data

2.1.1 Comparator Reduction

The first factor to be considered in dealing with measurements on a comparator is that the comparator itself is not perfect. Inaccuracy in scale and lack of perpendicularity between X and Y axes are the two most obvious sources of comparator error. Although more elaborate formulations have been tested, involving screw harmonics, the following has proven to give sufficient accuracy in compensating for comparator errors in the preliminary stage:

$$\begin{aligned} x &= x'\cdot s_x{}^* + y'\cdot s_y{}^*\cdot\sin\alpha^* \\ y &= y'\cdot s_y{}^* \end{aligned} \tag{1}$$

where x' and y' are the coordinates recorded by the comparator, $s_x{}^*$ and $s_y{}^*$ are the scale factors in X and Y, and α^* the difference between 90° and the true angle between the X and Y axes. The comparators are calibrated at regular intervals; the scale factors and the α^* are among the parameters determined by calibration.

2.1.2 Coordinate Systems

The measurements obtained are in the coordinate system of the comparator, which is arbitrary as far as the plate is concerned. For the first years of satellite triangulation work, all measurement data were transformed to a system determined by the plate fiducial marks or drill holes. For each subset, *i.e.*, the measurements done in a single session, a system was defined by the origin at the center of the drill holes, X-axis positive toward drill hole 3 and Y-axis positive toward drill hole 4. The mean of all opening and closing drill hole measurements for the subset was used.

In the 1968 reevaluation of the procedure, it was decided the data should be kept as close to the original as possible. Therefore the measurements are now retained in the comparator system except for a translation to the center of the drill holes. This center is computed as the least squares solution of the intersection of the lines defined by the four pairs of opposite drill holes. The observation equation is the two-point form of the equation of a straight line:

$$x[y(b)-y(a)] - y[x(b)-x(a)] = x(a)\cdot y(b) - x(b)\cdot y(a)$$

for each pair of opposite drill holes a and b. The normal equations are formed in the usual manner:

$$\begin{Vmatrix} \sum\,[y(b)-y(a)]^2 & \sum -[y(b)-y(a)]\cdot[x(b)-x(a)] \\ \sum -[y(b)-y(a)]\cdot[x(b)-x(a)] & \sum\,[x(b)-x(a)]^2 \end{Vmatrix} \cdot \begin{Vmatrix} x \\ y \end{Vmatrix} = \begin{Vmatrix} \sum\,[y(b)-y(a)]\cdot[x(a)y(b)-x(b)y(a)] \\ \sum -[x(b)-x(a)]\cdot[x(a)y(b)-x(b)y(a)] \end{Vmatrix} \tag{2}$$

The solution of the normals gives the coordinates of the center of the drill holes in the comparator coordinate system. All measurements including drill holes are then translated:

$$
\begin{aligned}
x_t &= x - x_c \\
y_t &= y - y_c
\end{aligned}
\tag{3}
$$

where x_c, y_c are the coordinates of the drill hole center, x,y indicate a measured point, and x_t, y_t are the coordinates of the point after translation. (All further equations deal with the translated measurements so the subscript t will be dropped.)

Since a considerable portion of the data processed to date has been in the fiducial system, and the possibility of rerunning any part of the processing must always be kept open, the programming must be kept flexible enough to handle data in either fiducial or comparator system. How this is done is explained in appendix A.

2.1.3 Possibility of Plate Upside Down

In future computations, the assumption is that the photographic plate is mounted in the comparator so that the image appears to the operator as it would in the idealized diapositive in figure 3. As can be seen from figure 4, this condition could be met either by viewing a negative plate with the emulsion side up, or a positive plate with the emulsion side down The latter is now standard, but as the method was being developed, all possible combinations were used. Furthermore, there is always the possibility of human error. For this reason, there must be a test and, if necessary, a correction, for this condition in the reduction program.

If the plate is mounted correctly, the drill hole numbers will increase in the counterclockwise direction. In the test for correct mounting, the drill hole measurements are transformed to the fiducial system as described in sec. 2.1.2, *i.e.*, besides the shift to center, they are rotated through τ:

$$
\tau = \tan^{-1}\left[\frac{y(3)-y(1)}{x(3)-x(1)}\right] \tag{4}
$$

This defines the positive direction of the X-axis (toward drill hole 3); the Y-axis retains the right-handed orientation (fig. 5a). In this coordinate system, the y-coordinate of drill holes 2 and 4 are computed:

$$
\begin{aligned}
y_T(2) &= (y(2)-y_C)\cos\tau - (x(2)-x_C)\sin\tau \\
y_T(4) &= (y(4)-y_C)\cos\tau - (x(4)-x_C)\sin\tau
\end{aligned}
\tag{5}
$$

If the plate was mounted correctly, $y_T(4)$ will be positive and $y_T(2)$ negative for counterclockwise drill holes (fig. 5b). If $y_T(4)$ is negative and $y_T(2)$ positive, it means that the drill holes are clockwise and so the plate was upside down (fig. 5c). In this case, the data may be converted to the diapositive mode by reversing the sign of the y-coordinates of all measurements after they have been translated to the center of the plate. The rotation of the drill hole measurements to the fiducial system is now done as a test only; the results are not retained.

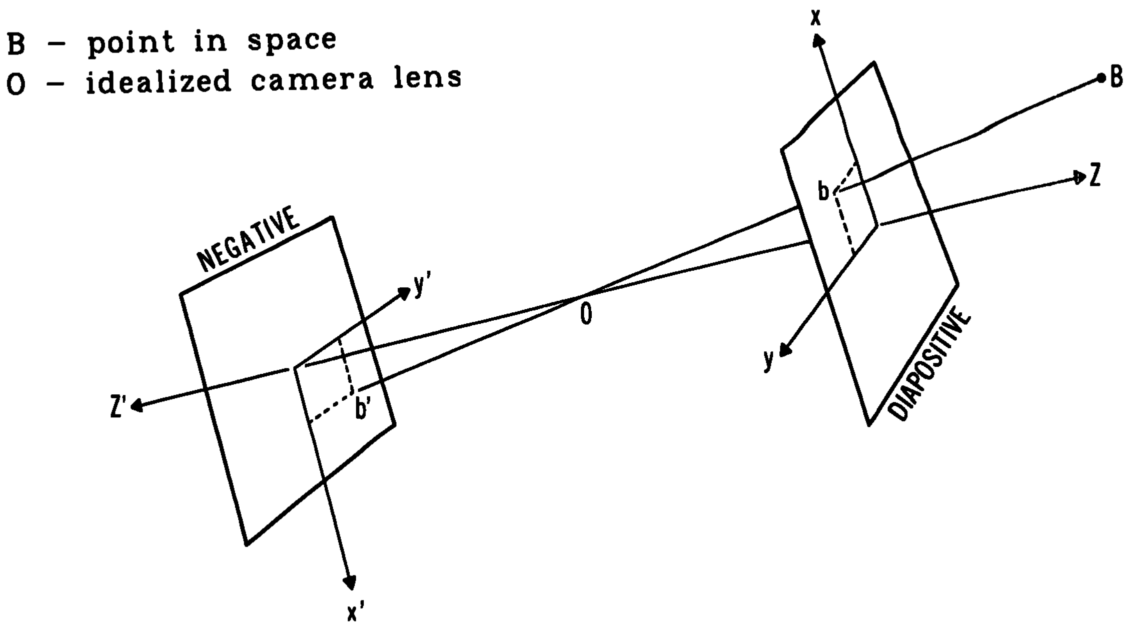
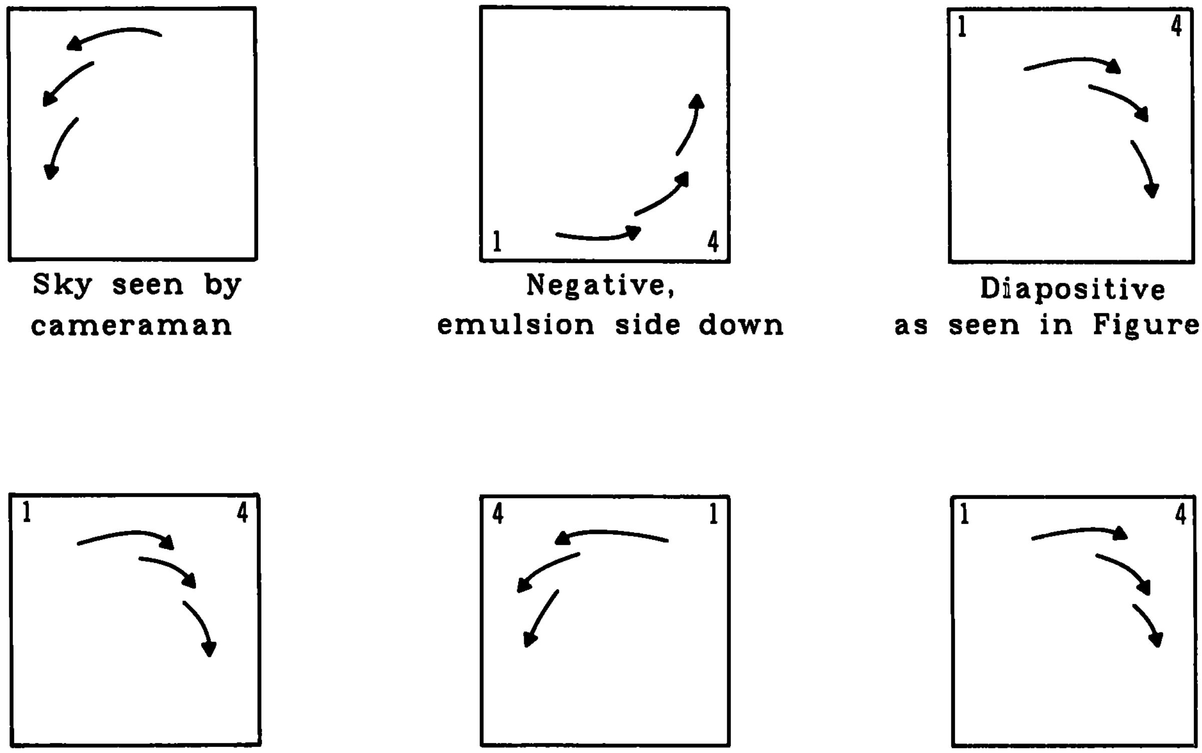


Figure 3. Relationship between negative and diapositive



Sky seen by cameraman

Negative, emulsion side down

Diapositive as seen in Figure 3

Negative, emulsion side up, rotated around horizontal axis

Positive, emulsion side up

Positive, emulsion side down

Figure 4. Relationship between star images under various conditions

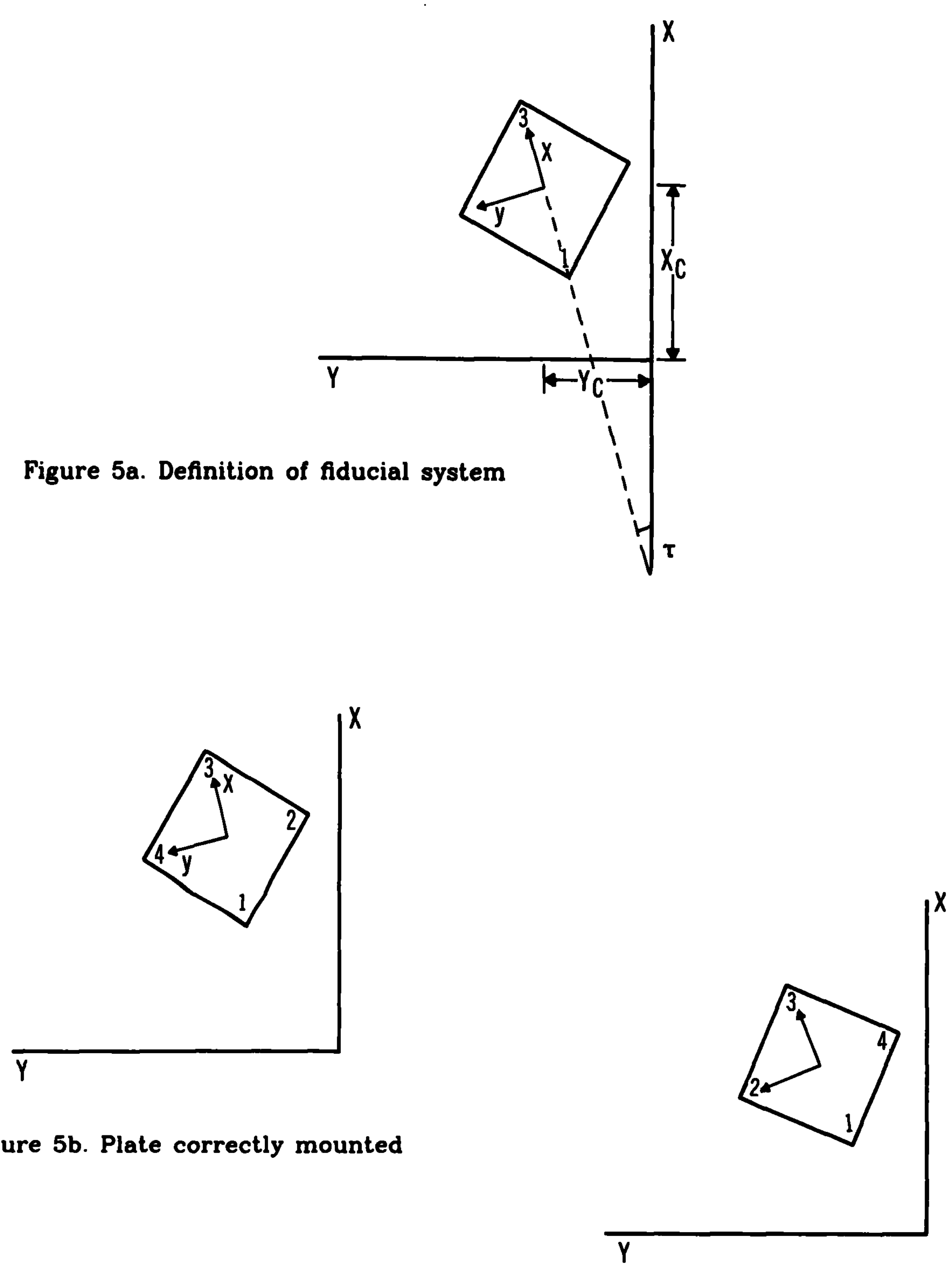


Figure 5a. Definition of fiducial system

Figure 5b. Plate correctly mounted

Figure 5c. Plate upside down

Figure 5. Plate mounted in comparator

2.2 Celestial Coordinates of the Stars

2.2.1 Source of Information

The source used for the stellar coordinates is the catalog published by the Smithsonian Astrophysical Observatory (1966). The contents of the catalog were studied by Bossler (1966); stars with visual magnitudes ≦8.0 and standard deviations ≦±0.4" at epoch 1950.0 were selected for use. After some minor editing to remove binaries, a total of 20,291 stars remained. For these stars, the right ascension (α), declination (δ), and the proper motions in each for epoch 1950.0 were stored on magnetic tape.

Because of long-term changes in the direction of the earth's axis, and motion of the stars themselves, their celestial coordinates will have changed between epoch 1950.0 and the time of observation. Thus it is necessary to update each star position for use in precision computations. It has been found convenient to perform the updating of the stars' positions in two parts, the first to the beginning of the Besselian year nearest the day of observation, the second to the time of observation itself.

2.2.2 Updating to Year of Observation

For the first phase, the method of updating used is that of Scott and Hughes (1964). The initial step is to transform the celestial coordinates for epoch and equinox 1950.0 to an earth-centered rectangular system, with the Z-axis toward the pole, the X-axis toward the vernal equinox, and the Y-axis lying in the equatorial plane and forming a right-handed system. In this system (with a celestial sphere of unit radius):

$$\begin{aligned} x &= \cos\alpha\cos\delta \\ y &= \sin\alpha\cos\delta \\ z &= \sin\delta. \end{aligned} \tag{6}$$

A comparable transformation is performed for the proper motions. Since $\mu_a = da$ where a is any direction and μ_a is the proper motion in that direction:

$$\begin{aligned} \mu_x = dx &= -\sin\alpha\cos\delta\, d\alpha - \cos\alpha\sin\delta\, d\delta \\ &= -\sin\alpha\cos\delta\,\mu_\alpha - \cos\alpha\sin\delta\,\mu_\delta. \end{aligned}$$

Similarly, (7)

$$\begin{aligned} \mu_y &= \cos\alpha\cos\delta\,\mu_\alpha - \sin\alpha\sin\delta\,\mu_\delta \\ \mu_z &= \cos\delta\,\mu_\delta. \end{aligned}$$

Next, updating to the epoch of the beginning of the Besselian year nearest the observation date is done:

$$\left\|\begin{matrix} x \\ y \\ z \end{matrix}\right\|_{0T} = \left\|\begin{matrix} x \\ y \\ z \end{matrix}\right\|_{00} + \left\|\begin{matrix} \mu_x & \dot{\mu}_x \\ \mu_y & \dot{\mu}_y \\ \mu_z & \dot{\mu}_z \end{matrix}\right\|_{00} \cdot \left\|\begin{matrix} \Delta T \\ \frac{1}{2}(\Delta T)^2 \end{matrix}\right\| \tag{8}$$

where subscript 00 means epoch and equinox of 1950, subscript $0T$ means epoch T referred to the equinox of 1950, and

$$\mu^2 = \mu_x{}^2 + \mu_y{}^2 + \mu_z{}^2$$

$$\dot{\mu}_x = -x\mu^2$$

$$\dot{\mu}_y = -y\mu^2$$

$$\dot{\mu}_z = -z\mu^2$$

T = nearest Besselian year

$$\Delta T = T - 1950.$$

Finally the transformation to the mean equinox T'

$$\left\| \begin{matrix} x \\ y \\ z \end{matrix} \right\|_{T'T} = \left\| \begin{matrix} X_x & Y_x & Z_x \\ X_y & Y_y & Z_y \\ X_z & Y_z & Z_z \end{matrix} \right\| \cdot \left\| \begin{matrix} x \\ y \\ z \end{matrix} \right\|_{0T} \tag{9}$$

where

$$T_0 = .5$$

$$\Delta_0'' = (2304.250+1.396T_0)\frac{\Delta T}{100} + .302\left(\frac{\Delta T}{100}\right)^2 + .018\left(\frac{\Delta T}{100}\right)^3$$

$$z'' = \Delta_0 + .791\left(\frac{\Delta T}{100}\right)^2$$

$$\theta'' = (2004.682 - .853T_0)\frac{\Delta T}{100} - .426\left(\frac{\Delta T}{100}\right)^2 - .042\left(\frac{\Delta T}{100}\right)^3$$

$$X_x = \cos\Delta_0 \cos\theta \cos z - \sin\Delta_0 \sin z$$

$$Y_x = -\sin\Delta_0 \cos\theta \cos z - \cos\Delta_0 \sin z$$

$$Z_x = -\sin\theta \cos z$$

$$X_y = \cos\Delta_0 \cos\theta \sin z + \sin\Delta_0 \cos z$$

$$Y_y = -\sin\Delta_0 \cos\theta \sin z + \cos\Delta_0 \cos z$$

$$Z_y = -\sin\theta \sin z$$

$$X_z = \cos\Delta_0 \sin\theta$$

$$Y_z = -\sin\Delta_0 \sin\theta$$

$$Z_z = \cos\theta.$$

This completes the updating to the beginning of the year.

2.2.3 Updating to Time of Observation

The second phase uses the method of Independent Day Numbers (sec. 1.3.4). The star coordinates for the beginning of the year closest to observation must be reconverted to right ascension and declination:

$$\alpha_0 = \tan^{-1}\left(\frac{y}{x}\right)$$
$$\delta_0 = \tan^{-1}\left(\frac{z}{\sqrt{x^2+y^2}}\right) \qquad (10)$$

where x, y, and z are the vector as above

$$\left\| \begin{matrix} x \\ y \\ z \end{matrix} \right\|_{T'T}$$

Then the updating formulation is:

$$\alpha = \alpha_0 + f + \tau\mu_\alpha + g\tan\delta_0\sin(\alpha_0+G) + \frac{h\sin(\alpha_0+H)}{\cos\delta_0} + J\tan^2\delta_0$$
$$\delta = \delta_0 + \tau\mu_\delta + h\sin\delta_0\cos(\alpha_0+H) + g\cos(\alpha_0+G) + i\cos\delta_0 + J'\tan\delta_0 \qquad (11)$$

Now α and δ represent the celestial coordinates of the star at the time of observation.

2.3 Adjustments to Celestial Coordinates

The coordinates of the star as computed in the previous section represent the true position of the star, but this is not the position as actually seen from earth. Because of diurnal aberration and refraction, the path of the light ray is deviated and falls on a slightly different place on the plate. It would be possible to adjust the measurements on the plate to conform with what they would be if these effects did not exist. But the philosophy has been to leave the measured data alone, and instead to compute what the star position would be, corresponding to the given measured position, in the absence of refraction and diurnal aberration.

For a detailed discussion of diurnal aberration and refraction, see Slama (1972). For diurnal aberration,the adjusted celestial coordinates are (in radians):

$$\mathrm{H} = \mathrm{ST} - \alpha$$
$$\alpha' = \alpha + \frac{1.55\times10^{-6}\cos\varphi\cos H}{\cos\delta} \qquad (12)$$
$$\delta' = \delta + 1.55\times10^{-6}\cos\varphi\sin H\sin\delta$$

where α and δ are the updated celestial coordinates and φ is the latitude of the observing station. ST is the local sidereal time of the observation.

The classical model of refraction has proved sufficiently accurate for the purposes of reduction of plate measurements. This model is represented as:

$$\begin{aligned} P' &= P/P_0 \quad \text{(in mm)} \\ T' &= T/T_0 \quad \text{(in °K)} \\ R &= P'/T'(C_0 \tan z + C_1 \tan^3 z + C_2 \tan^5 z) \end{aligned} \tag{13}$$

where R is in radians, P and T are the air pressure and temperature, P_0 and T_0 are standard pressure and temperature (P_0 = 760 mm, T_0 = 273°K), and z is the refracted zenith distance of the star. The values used for the refraction parameters are (in radians):

$$\begin{aligned} C_0 &= 2.9137566\times10^{-4} \\ C_1 &= -3.227865\times10^{-7} \\ C_2 &= 1.0225\times10^{-9} \end{aligned} \tag{14}$$

The pressure and temperature of the air are in practice known only at the station. An estimate of average air pressure is made as a function of the station's latitude (φ) and elevation (E) in meters:

$$P = P_{sta}\left(1 - .00264 \cos(2\varphi) - \frac{2E}{6.37\times10^6}\right) \tag{15}$$

Station temperature is assumed to be sufficiently accurate for this use.

Note also that when used in processing, the formula is set up so that temperatures may be input in degrees centigrade rather than Kelvin.

$$T^K/T_0^K = 1 + \beta T^C \qquad \text{where } \beta = 1/273 \tag{16}$$

The adjusted star positions are:

$$\begin{aligned} \delta'' &= \delta' + R\cos q \\ \alpha'' &= \alpha' + \frac{R\sin q}{\cos\delta''} \end{aligned} \tag{17}$$

where

$$\begin{aligned} \sin q &= \frac{\sin H' \cos\varphi}{\sin z} \qquad (H' = \mathrm{ST} - \alpha') \\ \cos q &= \frac{\sin\varphi - \sin\delta'\cos z}{\cos\delta'\sin z} \end{aligned} \tag{18}$$

The zenith distance of the star must be recomputed at each stage of correction:

$$z = \cos^{-1}(\sin\varphi\sin\delta + \cos\varphi\cos H\cos\delta). \tag{19}$$

2.4 Mathematical Representations of Star Position

2.4.1 Direction Cosine Matrix

See Slama (1972) for a detailed discussion of the following relations. In photogrammetry, the relation between the position of an object in space *XYZ* and its image on a photographic plate x,y may be computed, given six parameters of the camera's orientation. These parameters are α and ω, the angle of the camera axis relative to the Z-axis and to the XZ-plane, respectively; the angle κ representing plate rotation about the camera axis; x_p and y_p, the coordinates of the principal point on the plate; and c, the focal length of the camera.

The position of the object being photographed may be computed:

$$
\begin{aligned}
X-X_0 &= \frac{(Z-Z_0)[(x-x_p)A_1 + (y-y_p)A_2 + cD]}{(x-x_p)C_1 + (y-y_p)C_2 + cF} \\
Y-Y_0 &= \frac{(Z-Z_0)[(x-x_p)B_1 + (y-y_p)B_2 + cE]}{(x-x_p)C_1 + (y-y_p)C_2 + cF}
\end{aligned}
\tag{20}
$$

The reverse, computing plate coordinates from position in space:

$$
\begin{aligned}
x-x_p &= \frac{c[(X-X_0)A_1 + (Y-Y_0)B_1 + (Z-Z_0)C_1]}{(X-X_0)D + (Y-Y_0)E + (Z-Z_0)F} \\
y-y_p &= \frac{c[(X-X_0)A_2 + (Y-Y_0)B_2 + (Z-Z_0)C_2]}{(X-X_0)D + (Y-Y_0)E + (Z-Z_0)F}
\end{aligned}
\tag{21}
$$

In these equations, X_0 Y_0 Z_0 are the coordinates of the camera, and

$$
\begin{aligned}
A_1 &= -\cos\alpha\cos\kappa + \sin\alpha\sin\omega\sin\kappa \\
B_1 &= -\cos\omega\sin\kappa \\
C_1 &= \sin\alpha\cos\kappa + \cos\alpha\sin\omega\sin\kappa \\
A_2 &= -\cos\alpha\sin\kappa - \sin\alpha\sin\omega\cos\kappa \\
B_2 &= \cos\omega\cos\kappa \\
C_2 &= \sin\alpha\sin\kappa - \cos\alpha\sin\omega\cos\kappa \\
D &= \sin\alpha\cos\omega \\
E &= \sin\omega \\
F &= \cos\alpha\cos\omega.
\end{aligned}
\tag{22}
$$

These nine parameters are referred to as the direction cosine matrix.

2.4.2 Standard Coordinates

If stars are the objects being considered, their positions in terms of celestial coordinates must be transformed into rectangular coordinates to make use of the direction cosine matrix. Since it is actually the position of the celestial object relative to the camera that is computed, a convenient rectangular system is a projection of the celestial sphere onto a plane tangent at the observer's zenith. On the plane, the ξ-axis is positive toward the north and the η-axis is positive toward the east. The third axis is through the zenith and has the value of unity. This is called a system of standard coordinates. In this case, in the photogrammetric equation:

$$\begin{aligned} X - X_0 &= \xi \\ Y - Y_0 &= \eta \\ Z - Z_0 &= 1. \end{aligned} \tag{23}$$

2.4.3 Conversion from Celestial to Standard Coordinates and Vice Versa

For a given observation location and local sidereal time, the standard coordinates for a star may be computed from its celestial coordinates:

$$\begin{aligned} \xi &= \frac{\cos\varphi \sin\delta - \sin\varphi \cos H \cos\delta}{\sin\varphi \sin\delta + \cos\varphi \cos H \cos\delta} \\ \eta &= \frac{-\cos\delta \sin H}{\sin\varphi \sin\delta + \cos\varphi \cos H \cos\delta} \end{aligned} \tag{24}$$

where H = ST − α. (Note that the denominator in both equations is the cosine of the zenith distance.)

Computation of celestial coordinates from standard coordinates requires an intermediate step: computing azimuth and zenith distance, A and z.

$$\begin{aligned} A &= \tan^{-1}(\eta/\xi) + \pi \qquad \text{(from south)} \\ z &= \tan^{-1}[(\xi^2+\eta^2)^{\frac{1}{2}}]. \end{aligned} \tag{25}$$

Then

$$\begin{aligned} H &= \tan^{-1}\left[\frac{\sin A \sin z}{\cos\varphi \cos z + \sin\varphi \cos A \sin z}\right] \\ \alpha &= ST - H \\ \delta &= \sin^{-1}(\sin\varphi \cos z - \cos\varphi \cos A \sin z). \end{aligned} \tag{26}$$

(Note that celestial coordinates computed from standard coordinates that were computed from plate measurements, are "as seen," *i.e.*, they contain adjustments for refraction and diurnal aberration.)

2.5 Least Squares Solution

The method of least squares is employed at several places in the reduction procedure. This familiar tool for fitting experimental data to a mathematical expression called the observation equation need not be discussed in detail. But to employ this method it is necessary that the observation equation be linear. In reduction of the measurement data it is necessary to perform various coordinate transformations by least squares, and these equations are non-linear.

$$\begin{aligned} x' &= \;\;[(x-\Delta x)(1+S_x) + (y-\Delta y)(1+S_y)\alpha]\cos\kappa + (y-\Delta y)(1+S_y)\sin\kappa \\ y' &= -[(x-\Delta x)(1+S_x) + (y-\Delta y)(1+S_y)\alpha]\sin\kappa + (y-\Delta y)(1+S_y)\cos\kappa \end{aligned} \tag{27}$$

where

x,y = coordinates in the original system

x',y' = coordinates in the new system

S_x,S_y = scale factor differentials between the two systems

α = angle representing lack of perpendicularity in the original system (assumed small enough that $\sin\alpha = \alpha$)

$\Delta x,\Delta y$ = shift between the two systems

κ = rotation between the two systems.

S_x, S_y and/or α may be neglected if a less rigorous solution is needed.

The equations must be linearized using Taylor's series, neglecting second order:

$$\begin{aligned} x' &= x^0 + \frac{\partial x^0}{\partial \Delta x^0}\Delta(\Delta x)^0 + \frac{\partial x^0}{\partial \Delta y^0}\Delta(\Delta y)^0 + \frac{\partial x^0}{\partial S_x^{\,0}}\Delta S_x^{\,0} + \frac{\partial x^0}{\partial S_y^{\,0}}\Delta S_y^{\,0} + \frac{\partial x^0}{\partial \alpha^0}\Delta\alpha^0 + \frac{\partial x^0}{\partial \kappa^0}\Delta\kappa^0 \\ y' &= y^0 + \frac{\partial y^0}{\partial \Delta x^0}\Delta(\Delta x)^0 + \frac{\partial y^0}{\partial \Delta y^0}\Delta(\Delta y)^0 + \frac{\partial y^0}{\partial S_x^{\,0}}\Delta S_x^{\,0} + \frac{\partial y^0}{\partial S_y^{\,0}}\Delta S_y^{\,0} + \frac{\partial y^0}{\partial \alpha^0}\Delta\alpha^0 + \frac{\partial y^0}{\partial \kappa^0}\Delta\kappa^0 \end{aligned} \tag{28}$$

The partials are evaluated:

$$\begin{array}{ll} \dfrac{\partial x}{\partial \Delta x} = -(1+S_x)\cos\kappa & \dfrac{\partial y}{\partial \Delta x} = \;\;(1+S_x)\sin\kappa \\ \dfrac{\partial x}{\partial \Delta y} = -(1+S_y)(\alpha\cos\kappa + \sin\kappa) & \dfrac{\partial y}{\partial \Delta y} = -(1+S_y)(-\alpha\sin\kappa + \cos\kappa) \\ \dfrac{\partial x}{\partial S_x} = \;\;(x-\Delta x)\cos\kappa & \dfrac{\partial y}{\partial S_x} = -(x-\Delta x)\sin\kappa \\ \dfrac{\partial x}{\partial S_y} = \;\;(y-\Delta y)(\alpha\cos\kappa + \sin\kappa) & \dfrac{\partial y}{\partial S_y} = \;\;(y-\Delta y)(-\alpha\sin\kappa + \cos\kappa) \\ \dfrac{\partial x}{\partial \alpha} = \;\;(y-\Delta y)(1+S_y)\cos\kappa & \dfrac{\partial y}{\partial \alpha} = -(y-\Delta y)(1+S_y)\sin\kappa \\ \multicolumn{2}{l}{\dfrac{\partial x}{\partial \kappa} = -((x-\Delta x)(1+S_x) + (y-\Delta y)(1+S_y)\alpha)\sin\kappa + (y-\Delta y)(1+S_y)\cos\kappa} \\ \multicolumn{2}{l}{\dfrac{\partial y}{\partial \kappa} = -((x-\Delta x)(1+S_x) + (y-\Delta y)(1+S_y)\alpha)\cos\kappa - (y-\Delta y)(1+S_y)\sin\kappa} \end{array} \tag{29}$$

An iterative method is then employed. In the first approximation, x^0 and y^0 are the coordinates in the original system and the parameters are zero (unless there is advance information about the rotation angle κ). The solution yields values, not for the parameters themselves, but for incremental changes to the assumed values of the parameters. After each iteration new values for x^0 and y^0 are computed using the original observation equations. The solution is then repeated using these new values and the parameters that are the cumulative result of previous iterations. The process continues until the computed values of x^0 and y^0 are within a tolerance limit of the observed values:

$$\frac{|\sum (x'-x^0)| + |\sum (y'-y^0)|}{N} < 10^{-8} \tag{30}$$

(In case of nonconvergence there is a limit of 10 iterations.)

Then the parameters used during the last iteration are taken as the final values.

In all cases where the least squares method is employed, the method for inverting the normals is one devised by Erwin Schmid (1973).

3. THE PRE SET OF MEASUREMENTS

3.1 Purpose

The functions of the "Pre set" are: to provide the framework by which the star and satellite images in the two main sets may be numbered and identified, and to provide a first approximation to the camera orientation.

The satellite images in the Pre set were numbered by the operator before measuring, according to their sequence position. These preidentified images were selected at roughly equal intervals along the trail; all the remaining images in the trail will be numbered accordingly.

The preidentified stars are the basis for the other functions of the Pre set. The identity and celestial coordinates of these stars are known; using the plate constant method, an approximation to the camera orientation is computed. This approximate camera orientation will be used in three places during the data processing.

a. Together with the station latitude, it will be used to determine the apparent direction of the stars' path across the plate. The star images, which had only their location code and trail number assigned during measurement, will have their sequence numbers computed according to this direction.

b. It will be used to compute an approximate right ascension and declination for each star image, which in turn will be used as the basis for a search through the star catalog to find the actual identity and celestial coordinates of the star.

c. It will serve as the first approximation to camera orientation for the Single Camera Orientation program, which uses the successive iteration method to compute its results (Slama 1972).

3.2 Input to Preliminary Reduction Program

The Pre set, after conversion from paper tape to cards, typically consists of:

a. eight drill holes, class 3, numbered 100 to 800 – up to five measurements each;

b. seven stars, class 2, numbered with five-digit star code, the last two digits of which are 0 – five measurements each (representing five images of a trail);

c. five satellites, class 9, numbered with sequence number – one measurement each.

To this set is now added:

d. station latitude and longitude,

e. comparator calibration parameters,

f. Independent Day Number parameters for the midpoint of each star trail (sec. 1.3.4),

g. celestial coordinates of preidentified stars (sec. 1.3.2), and

h. descriptive information.

This forms the input to the Preliminary Reduction program, designated program 673, written for the CDC 6600.

3.3 Operations of the Preliminary Reduction Program

3.3.1 Initial Steps

For each drill hole, a mean of the several measurements is computed to obtain the best value for the positions. A mean of the five measurements for each star is computed even though they represent five different images in a trail. This value is assumed to approximate the midpoint of the trail as accurately as necessary for the purposes of the program. The meaned measurements of drill holes and stars, and the single measurements of satellites, are subjected to comparator reduction, translation to plate center, and test for plate mounting [eq.(1) to (5)].

3.3.2 Celestial Coordinates and Standard Coordinates

At the time the star chart program produced the chart for the particular plate, a card was punched for each possible preidentified star with its location code and catalog information, including right ascension, declination, and proper motions in each, for the epoch of 1950.0. When certain of these stars were selected for measurement, their location codes were incorporated in their image numbers. Now, to associate each measured position with the appropriate celestial coordinates, the program searches through the list of star positions (input by means of cards) and matches location codes.

As each star position is matched, the celestial coordinates α, δ are updated. The first step updates to the beginning of the Besselian year nearest the observation [eq.(6) to (9)].

To update stellar coordinates from the beginning of the year to the time of observation, the Independent Day Number parameters for the exposure time of the image in question must be known. The parameters for the midpoint of each trail have been read into the program. The routine matches each meaned star image with the appropriate parameters by matching trail numbers. The updating of the stellar coordinates continues as in eq.(10) and (11), neglecting the terms in J and J' since they are second-order corrections.

The position of each star after updating must be adjusted for the effects of diurnal aberration and refraction [eq.(12) to (19)]. Then the standard coordinates ξ,η for each star position are computed from the updated and corrected celestial coordinates α,δ [eq.(24)].

3.3.3 Computation of Camera Orientation

After the above operations are performed for all the measured, preidentified star images, the plate constant method is employed to compute the camera orientation. This method is based on the pair of observation equations relating star position ξ,η and associated plate position x,y:

$$\begin{aligned} \xi_i A1 + \eta_i B1 + C1 - \xi_i x_i A0 - \eta_i x_i B0 &= -x_i \\ \xi_i A2 + \eta_i B2 + C2 - \xi_i y_i A0 - \eta_i y_i B0 &= -y_i \end{aligned} \qquad (31)$$

These equations are linear and the least squares method may be applied directly. The normal equations are formed in the usual way. Solution, by inversion of the matrix and multiplication by the column on the right, results in best fit values for the plate constants $A1$ through $B0$. The camera parameters (see sec. 2.4.1) may be computed from the plate constants:

$$\begin{aligned}
\alpha &= \tan^{-1}(A0) \\
\omega &= \frac{\tan^{-1}(B0)}{[(A0)^2 + 1]^{1/2}} \\
\kappa &= \tan^{-1}\left(\frac{A0\,C2\ -\ A2}{A0\,C1\ -\ A1}\right) \\
x_p &= \frac{A0\,A1\ +\ B0\,B1\ +\ C1}{(A0)^2\ +\ (B0)^2\ +\ 1} \\
y_p &= \frac{A0\,A2\ +\ B0\,B2\ +\ C2}{(A0)^2\ +\ (B0)^2\ +\ 1} \\
c &= \left[\frac{\left(1 + \frac{A2\,B0 - A0\,B2}{A1\,B2 - A2\,B1}x_p + \frac{A0\,B1\ - A1\,B0}{A1\,B2 - A2\,B1}y_p\right)^3\left(A1\,B2 - A2\,B1\right)^2}{1 + \frac{A2\,B0 - A0\,B2}{A1\,B2 - A2\,B1}\,C1 + \frac{A0\,B1\ - A1\,B0}{A1\,B2 - A2\,B1}\,C2}\right]^{\frac{1}{4}}
\end{aligned} \tag{32}$$

3.3.4 Test of Solution

It is possible that at least one of the preidentified stars was wrongly identified or badly measured, thus throwing the solution off. If such stars can be pinpointed, they can be eliminated from the solution and the parameters recomputed. Seven stars are usually selected as preidentified. Four stars would provide a unique solution; therefore, five stars is the minimum accepted by the program.

The first step in testing the solution is to compute the standard coordinates of the stars from their plate measurements and the computed orientation parameters using eq.(20), (22), and (23).

$$\begin{aligned}
\xi^c &= \frac{(x-x_p)A_1 + (y-y_p)A_2 + cD}{(x-x_p)C_1 + (y-y_p)C_2 + cF} \\
\eta^c &= \frac{(x-x_p)B_1 + (y-y_p)B_2 + cE}{(x-x_p)C_1 + (y-y_p)C_2 + cF}
\end{aligned} \tag{33}$$

where superscript c means computed. The residuals between these computed values and the values used in finding the solution are computed for each star:

$$\begin{aligned}
v_\xi &= \xi - \xi^c \\
v_\eta &= \eta - \eta^c
\end{aligned} \tag{34}$$

These residuals are then compared with a tolerance, usually 100μm. If all residuals are less than this tolerance, the solution is considered satisfactory. If one star has residual(s) greater than the tolerance, the star is removed and the solution repeated, using eq.(31) to (34). If more than one star has residuals greater than the tolerance, it is assumed that the solution is sufficiently bad so the star causing the trouble cannot be identified by residuals. In this case, the solution is repeated automatically, removing each star in turn until a satisfactory solution is reached. If none of these methods succeeds in obtaining a solution with more than four stars, the Pre set must be rejected, and different stars selected.

3.3.5 Output

The following information is output from the program:

a. eight drill holes, adjusted for comparator;

b. five numbered satellite images, adjusted for comparator;

c. cards to be used in Single Camera Orientation program, including descriptive information, camera orientation parameters, station coordinates; and

d. cards to be used in Plate Data Reduction program, including comparator parameters, camera orientation parameters, station coordinates.

This information is written on a tape in card image form, and is later punched from the tape for use in appropriate places in the remaining procedure.

4. REDUCTION OF THE MAIN DATA SETS

4.1 Introduction

The reduction of the data in the two main sets of plate measurements is the heart of the data preparation procedure. Its goal is to produce a set of plate coordinates, with comparator and operator biases removed, for star and satellite images in a single consistent plate-centered system, with each image readily associated with the time of its exposure. To do this, the data must be subjected to the operations:

a. Comparator reduction, translation to center, test for plate mounting;
b. Patching, to put subsets of data measured at different sessions into one consistent system;
c. Star and satellite image numbering, using information from the Preliminary Reduction program;
d. Matching, to combine the two sets of measurements made 180° apart, based on star images only;
e. Satellite transformation, to apply the results of matching to the satellite images; and
f. Statistical analysis of the results of the various operations.

4.2 Input

It will be recalled that each of the two main data sets, A and B, is composed of several subsets, each representing data measured in different sessions. Each subset typically consists of:

a. four pre-session drill holes, class 0, numbered 100 to 400 – five measurements each;
b. four post-session drill holes, class 0, numbered 100 to 400 – five measurements each;
c. eight drill holes, pre- and post-session combined, class 3, numbered 100 to 800 – ten measurements each;
d. star images, class 2, numbered with location code and trail number – five measurements each (representing five images of a trail); and
e. satellite images, class 9, no image number – one measurement each.

The format of these data is:

code no. for plate	class	image number	x-coord. (m)	y-coord. (m)
I4,	I1,	I3,	F6.6,	F6.6

The subsets are assembled with the B set first, followed by the A set. The end of each set is signaled by a 1-punch in cc. 79 after the B, and a 1-punch in cc. 80 after the A. These data are now preceded by:

a. eight Pre set drill holes and five numbered satellite images, from Preliminary Reduction program;

b. comparator parameters, approximate camera orientation, and station position, from Preliminary Reduction program;
c. tabulation of exposure pattern, from Independent Day Number program; and
d. tolerances allowed for patching, matching, and star image numbering.
This forms the standard input to program 383, Plate Data Reduction, written for the CDC 6600.

4.3 Test for Drill Hole Closure

When the input data are read in, the plate measurements are stored on file TAPE 1 on the computer's storage disk. In addition, the class 0 and class 3 points are used to test the quality of the measuring during each subset by computing the amount of change between the measured positions of the drill holes at the beginning and end of the session. When, in 1968, the necessity of this test became apparent and all the plates were to be reprocessed, it was necessary to find a way to distinguish between opening and closing measurements of the drill holes of these plates. The actual mechanics of this method, which occupy most of subroutine DRHCHK, are tedious and need not be described in detail. Class 0 points are now used solely for this purpose (see sec. 1.3.3).

When the measurements of a given drill hole have been separated into opening and closing, the mean x- and y-positions for opening and closing are computed, and then the difference between the two positions is computed. When all eight drill holes have been processed this way, the differences are summed. If this sum, called the closure, is greater than 11μm, or if any drill hole has a difference between opening and closing of more than 3.5μm, the subset is considered unacceptable and the plate must be remeasured.

After passing through the drill hole closure test, the mean of all positions of each drill hole is computed. The comparator reduction operation is then applied to each meaned drill hole position [eq.(1)].

4.4 Patching

Subsets of data measured at different sessions must be put into one consistent system by relating their drill hole positions. Within each of the A and B sets, the subset with the smallest drill hole closure is taken as the standard and the other subsets are transformed to its system. Normally, it is expected that differences between the subsets will be quite small, since minor temperature–humidity fluctuations in the comparator area are the only source of change. (See appendix A for consideration of large shifts between subsets.)

For each subset of drill holes, a least squares fit is made to the drill holes of the standard subset for A or B. The fit is based on the equations:

$$\begin{aligned} x' &= (x-\Delta x)(1+S_x)\cos\kappa + (y-\Delta y)(1+S_y)\sin\kappa \\ y' &= -(x-\Delta x)(1+S_x)\sin\kappa + (y-\Delta y)(1+S_y)\cos\kappa \end{aligned} \qquad (35)$$

which are the transformation equations eq.(27), neglecting α. Since this transformation assumes that rotation κ is relative to the center of the final coordinate system, the results will be incorrect unless all measurements currently in the comparator system are first shifted to the center of the standard subset of drill holes. This in turn would cause theoretical difficulties because the final plate center to which the star and satellite images will be translated prior to their transformation by the patching parameters is not necessarily the same as the center of the standard subset to which the patching parameters are being referenced. In practice, the difficulty vanishes because of the usually small differences between plate positions for successive subsets. In fact, it is necessary only to shift the drill hole measurements approximately to the center of the standard subset, so the first subset of the A and the B is used for convenience, [eq.(2) and (3)].

Because only eight points are involved, it is possible for one bad measurement to have a large adverse effect on the patching results. For this reason, a pre-patching test is applied. The difference between the position of each drill hole in the standard subset and other subsets is computed and checked against a tolerance. This tolerance is part of input so may be changed for differing situations; the usual value is 30μm. If the difference is greater than this, the drill hole is not used in the solution.

The method of solution using least squares is the iterative method, [eq.(28) to (30)]. After the five transformation parameters for each subset are computed, they are stored in memory so they may be applied to all measurements within the subset. When all subsets are completed, the drill holes of each in turn are transformed in this way [eq.(35)]. In effect, we now have several independent measurements in a consistent system for each drill hole. These "patched" drill holes are meaned for sets A and B to give the final values.

4.5 Star and Satellite Images

4.5.1 Plate Center and Test for Plate Mounting

The computation of the center of the meaned drill holes is performed to give a final value for the center of the plate for sets A and B by eq.(2). This will be the reference point to which all star and satellite measurements will be shifted. The test for plate mounting is performed [eq.(4) and (5)], and the signal retained, if necessary, for reversing all y-coordinates.

4.5.2 Preparation for Satellite Numbering

The prenumbered satellite images from the Preliminary Reduction program were measured in the Pre set; to use them for numbering the images of the A and B sets, their measured positions must be transformed in turn to the two systems determined by the A and B drill holes. A least squares computation similar to that of patching is performed for each case, using only Δx, Δy, and κ as parameters:

$$\begin{aligned} x' &= (x-\Delta x)\cos\kappa + (y-\Delta y)\sin\kappa \\ y' &= -(x-\Delta x)\sin\kappa + (y-\Delta y)\cos\kappa. \end{aligned} \tag{36}$$

For the B set, the first approximation to κ is taken as 180°; for the A set, it is zero. The resulting values for the parameters are applied to the plate positions of the prenumbered satellite images, and the transformed positions stored in memory.

In the normal case the transformed values for these plate positions will differ very little from the original values, or, in the B set, from a 180° reversal of the original values. But the operation is done to allow for the possibility of the plate having been removed from the comparator between the Pre and the A sets, or an error in turning the plate 180° between the A and B sets.

4.5.3 Reduction of Star and Satellite Image Measurements

The data for the stars and satellites are now read in from peripheral storage one image at a time. Each measured position is put through the operations of comparator reduction [eq.(1)], shift to center [eq.(3)], patching to standard subset [eq.(35]), and, if necessary, y-coordinate reversal.

4.5.4 Satellite Numbering

The satellite images are to be numbered according to their relation in the satellite trail to the five pre-numbered images. The procedure for numbering was devised by Erwin Schmid (1965a).

Given: n point numbers P_i $(i=1,2,\ldots,n)$ and the corresponding plate coordinates (x_i, y_i). P_1 and P_n are end points of the trail segment. $P_2, P_3, \ldots P_{n-1}$ are chosen to make the $(n-1)$ portions of the segment roughly equal.

Find: point number P of a point with given (x,y).

$$D_i = [(x_i-x_1)^2 + (y_i-y_1)^2]^{\frac{1}{2}} \qquad \text{note } D_1 = 0$$

$$D = [(x-x_1)^2 + (y-y_1)^2]^{\frac{1}{2}}$$

$$\begin{aligned} P = {} & P_1\frac{(D-D_2)(D-D_3)(D-D_4)\ldots(D-D_n)}{(D_1-D_2)(D_1-D_3)(D_1-D_4)\ldots(D_1-D_n)} + P_2\frac{(D-D_1)(D-D_3)(D-D_4)\ldots(D-D_n)}{(D_2-D_1)(D_2-D_3)(D_2-D_4)\ldots(D_2-D_n)} \\ & + P_3\frac{(D-D_1)(D-D_2)(D-D_4)\ldots(D-D_n)}{(D_3-D_1)(D_3-D_2)(D_3-D_4)\ldots(D_3-D_n)} + \ldots + P_n\frac{(D-D_1)(D-D_2)(D-D_3)\ldots(D-D_{n-1})}{(D_n-D_1)(D_n-D_2)(D_n-D_3)\ldots(D_n-D_{n-1})} \end{aligned} \qquad (37)$$

The integer number nearest P is used as the point number. P as computed is also printed out to two decimal places as a check. This number should be within 0.1 of the integer value, since regular spacing of the satellite image points can be assumed because of the high accuracy of the shutter of the BC-4 camera. If the satellite numbers do not compute close to integer values, it is an indication that there is an error in the prenumbered satellites, or in their transformation to the coordinate system of the data. (See appendix B for procedure used with cameras other than BC-4.)

4.5.5 Star Numbering

The star images must have the rightmost two digits of their numbers assigned in accordance with the time-based numbering scheme. At one time, the comparator operator assigned the complete number to each point during the measurement phase. But this was found to be a source of frequent errors, especially in the last two digits which must run in chronological sequence. Plates exposed to regions near the pole presented a particularly troublesome problem, since the direction of increasing time is different for every part of the plate.

Under the present procedure, each trail to be measured is marked and its trail number noted in the pre-measuring phase. The operator, while measuring, causes the block number (visible on the plate) and the trail number to be punched in the paper tape with each measurement record. After the exposure pattern is tabulated and the sequence numbers associated with each trail are input to the Plate Data Reduction program, the numbering can be completed automatically.

For each star image the following series of operations are performed. Standard coordinates ξ,η are computed from the measured plate position, using the approximate camera orientation parameters [eq.(20),(22), and (23)]. The hour angle and declination are computed from the standard coordinates, using the azimuth and zenith distance [eq.(25) and (26)]. Since the star position as computed from plate measurements is, in effect, adjusted for diurnal aberration, this adjustment must be removed to get to the star's true position. Refraction is neglected here because only the relative position of the images is of concern. The correction is based on eq.(12).

$$H' = H + \frac{1.55\times10^{-6}\cos\varphi\cos H}{\cos\delta}$$

When all the images for one star have been processed in this way, the images are sorted on increasing hour angle. In this way, they are arranged, by definition, in the direction of increasing time, whatever their appearance on the plate. The initial sequence number for the particular trail is assigned to the earliest image, and the remaining images are numbered sequentially. The computed hour angle and the various intermediate steps are not retained.

There still remains the possibility of error in the location code or trail number as punched onto tape by the comparator operator. See sec. 4.6.4 for how this is handled.

4.5.6 Output Numbered Raw Data

The patched and numbered star and satellite images have been stored on file TAPE 3 on the disk. At this point in the program, the raw measurements are to be output in a form identical to the input except that the star and satellite images are completely numbered. This will be regarded as raw data for storage and rerun purposes (see Appendix A). The two sets of data stored on the disk, the original input version, and the patched and numbered version, are read into core and matched up. The image numbers from the latter are added to the former, and the results written on the output tape.

4.6 Matching

4.6.1 Separation of Star and Satellite Images

Because of the different nature of the celestial bodies producing the images, and because exposure times are different, it is probable that the star and satellite images have a different appearance on the plate. Therefore, it must be assumed that the comparator operator's measuring bias was different for the two types of images.

A rigorous solution would call for the simultaneous adjustment of star and satellite images, with different biases allowed for in the equation. However, it was felt that a satisfactory solution could be reached by adjusting the stars only, then applying the results to the satellites.

4.6.2 Theory of the Matching Operation

Since a perfect physical rotation of the plate between the A and B sets is not attainable in practice, a transformation between the two sets of measurements must be assumed. All six parameters of eq.(27) are used. Here κ is the difference between true rotation and 180°. Since the two sets of measurements were each shifted to the center of the plate as determined from the drill holes of that set, the real significance of the Δx and Δy in this formulation is the operator bias.

To eliminate the rotational effects, one set is reversed and a least-squares fit is made of the entire collection of measured star images between that set and the other. After this transformation we will have two sets of measurements in the same framework. These are meaned to give the best value for the positions of the images.

4.6.3 Choice of Set to be Taken as Standard

This choice was originally made on the assumption that the A set, the first measured, was better; the program requires the B set to be input first because, originally, it was always the secondary set. Then a study was done which showed that, on the contrary, the B set was often better (as shown by smaller drill hole closures), presumably because the B is often done in the afternoon when the comparator has warmed up and stabilized.

Now the program has been made flexible, so that whichever set is better may be used as standard. The drill hole closures of the best subsets of the A and B sets (those subsets used as the standard for patching) are compared. If the best subset of the A set has the smaller closure, the A is used as the standard for matching; if the B set has the smaller closure, it is used. The plate coordinates of all images in the secondary set are reversed.

4.6.4 Correction of Image Numbers

Images from the two sets are paired off by their image numbers. When all pairs have been located, there may remain star images in either or both sets which have no pairs. This is due to one of two reasons:

a. The comparator operator neglected to measure a marked trail in one of the two sets; or

b. The operator made an error in either the block or trail number in one set, so the trail was misnumbered.

In the latter case there is a possibility of retrieval, since the data actually are present in both sets. For each unpaired image in the primary set, a search is made through the unpaired images of the secondary set, comparing plate positions only. If these are within a certain tolerance, the images are assumed to be the same, and the number on the secondary set is changed to that of the primary set. Of course, there is a 50% chance that this is the wrong image number; if so, the error will be caught at a later stage of the processing.

The choice of tolerance is a delicate one; if the A and B sets are not close to a perfect 180° reversal some margin must be allowed. but if the tolerance is too large, an image may be paired with the wrong image in the trail or even an image of another star. Such an error would be caught later but the operation would be wasted. Under normal circumstances the error tolerance is 50μm.

4.6.5 Pre-solution Data Check

Before each star image is added to the equation, a pre-matching test is applied to weed out gross errors. The difference in position between corresponding images in the two sets must be less than a given tolerance. This is often 100μm, but it may be changed for specific situations.

4.6.6 Least Squares Solution

The equations are expanded in a Taylor's series [eq.(28) and (29)], and normals set up. The solution is repeated until the convergence test [eq.(30)] is met. All the images of the secondary set are transformed by the computed parameters [eq.(27)]. The differences between these and the primary set are studied. If any image has a difference greater than tolerances $T_x.T_y$, the image will be rejected and the solution rerun. The tolerances are:

$$T_x = 3\left(\frac{\sum d_x^2}{N-6}\right)^{\frac{1}{2}} \qquad T_y = 3\left(\frac{\sum d_y^2}{N-6}\right)^{\frac{1}{2}}$$

where d_x and d_y are the differences in x and y. This check serves to eliminate badly measured points not caught by the pre-matching check, and to reverse the effects of a too-large tolerance in the image renumbering

The solution is completely rerun without the rejected images. The results of the matching operation are obtained by meaning the primary set of measurements with the transformed secondary set. These are the final values of the star positions that will be output at the end of the program.

4.6.7 Statistical Analysis

The following are printed out for visual inspection:

a. computed parameters,
b. inverse of normals,
c. mean errors of position,

$$s_x^M = \frac{1}{2}\left(\frac{\sum d_x^2}{N-6}\right)^{\frac{1}{2}}$$

$$s_y^M = \frac{1}{2}\left(\frac{\sum d_y^2}{N-6}\right)^{\frac{1}{2}}$$

$$s^M = \frac{1}{2}\left(\frac{\sum d_x^2+\sum d_y^2}{2N-6}\right)^{\frac{1}{2}} \tag{38}$$

(Note: these are just half of the mean errors based on the transformation, because the error is divided between the two measurements of each position by taking the final position as the mean of the two.)

d. mean errors of computed parameters

$$s_i = s^M(F_{ii})^{\frac{1}{2}}$$

where F_{ii} is the diagonal term of the inverse of the normals, and
e. differences for all images, in micrometers.

The program will proceed, but, later, human judgment may be that the mean error of matching is too high (anything over 2.5μm is considered high). Decisions will then be made as to whether the plate should be remeasured, or the data rerun with certain points rejected in advance (see appendix A), or if special circumstances explain a high mean error without prejudicing the data.

4.6.8 Graphical Analysis of Differences

The differences in x and in y are now regarded as two distributions of numbers and are subjected to histogram analysis. These distributions should approximate the normal, because after the measurements of one set have been transformed to the other, the remaining differences should represent random noise only. To facilitate the comparison, a Gaussian normal curve is superimposed on the histogram.

The logic for the histogram routine was developed by Erwin Schmid (1965b) when limited computer facilities made it desirable that the same normal curve be used for all problems. A standard curve with unit area was used:

$$\varphi(t) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}t^2}. \tag{39}$$

To superimpose this curve on the histogram, the latter must be computed with the class interval, on the t-axis, in multiples of the standard deviation of the distribution, and the height, on the φ-axis, reduced in scale to make the total histogram area equal to unity. If Δ is class width and m the number of elements in a bar:

area of each bar $= m\Delta$

total area of histogram $= \sum m\Delta = \Delta\sum m = \Delta M$

height scale factor $= \dfrac{1}{\Delta M}$

scaled area of histogram $= \sum \dfrac{m}{\Delta M}\Delta = \dfrac{1}{M}\sum m = \dfrac{M}{M} = 1$

The disadvantage of this procedure is that the resulting coordinates on the φ-axis are very small. For convenience sake, all φ-coordinates for both histogram and normal curve are multiplied by a scale factor of 10.

In the Erwin Schmid method, the class width is always the reciprocal of an integer, N, so the outer boundaries are adjustable and the choice of number of classes limited. The mean of the distribution is assumed to be zero.

class width $\Delta = \dfrac{1}{N}$

number of classes $= 6N+1$

outer boundaries $= \pm\dfrac{6N+1}{2N}$

The boundaries between classes can be computed as:

$$-\left(\frac{6N+1}{2N}\right)+\frac{j}{N} \qquad \text{where } j = 1,2,....6N+1$$

In practice, N is always 2, so the number of classes is always 13 and the outer limits are ±3.25. Any values falling outside these limits are counted in the outer classes. A more flexible histogram routine was not considered necessary for this purpose.

The actual plotting is done on an off-line CRT device, the FR-80, from whence it is transferred to microfilm together with the printed output.

4.7 Satellite Transformation

4.7.1 Operation

Rotation, scale factors, and lack of perpendicularity are all related to the plate itself, or to the comparator, and are therefore the same for the satellite images as for the stars. The bias, however, is different. If the transformation parameters from matching are applied to the secondary set of satellite measurements by eq.(27), the remaining differences between these and the primary set will consist of bias plus random noise. Taking the mean of the primary and the transformed secondary sets eliminates the bias. The final results for the satellite images consist of these meaned positions.

4.7.2 Statistical Analysis

The following are printed out for visual inspection:

a. differences for all images, in micrometers;

b. mean of differences in x and y = bias;

c. mean error of transformation

$$s_x^T = \left(\frac{\sum(d_x-\bar{d}_x)^2}{4N}\right)^{\frac{1}{2}}$$

$$s_y^T = \left(\frac{\sum(d_y-\bar{d}_y)^2}{4N}\right)^{\frac{1}{2}}$$

$$s^T = \left(\frac{\sum(d_x-\bar{d}_x)^2 + \sum(d_y-\bar{d}_y)^2}{8N}\right)^{\frac{1}{2}} \qquad (40)$$

where $\bar{d}_x$ and $\bar{d}_y$ are the biases in x and y.

4.8 Curve Fit of Satellite Path

4.8.1 General Discussion

The fitting of a mathematical curve to the path of the satellite, and statistical analysis of the results, is a useful tool in evaluation of the data. The curve fit is not made to the satellite's path in space, but to the x- and y-coordinates in turn, versus time as represented by the image number. Since the satellite's path must be smooth in time, the closeness of the measured points to a mathematical curve is a test of the photographic, measurement, and reduction procedures.

Furthermore, it is not the coordinates of the data as measured (ultimately derived from the plate's orientation in the comparator) that are used, for these have no functional relationship to the satellite path. For the curve fit, the coordinates are rotated so that the trail of satellite images lies generally parallel to the X-axis.

Extensive study was carried out as to what form of mathematical curve should be used. Polynomials of various orders, as well as combinations of polynomial and harmonic terms, were tested. It was decided that the fifth-order polynomial is the optimum form. Another order may be substituted if desired.

4.8.2 Determination of Center

The center image number, which will become the zero-point of the time dimension, is determined by

$$N_c = \frac{N_1 + N_n}{2}.$$

4.8.3 Rotation of Coordinates

All measured points are rotated to a coordinate system in which the direction of the X-axis is defined by the two endpoints of the satellite trail (see eq.(4)).

$$\tau = \tan^{-1}\left(\frac{y_1 - y_n}{x_1 - x_n}\right)$$

$$x^R = x \cos\tau + y \sin\tau$$

$$y^R = y \cos\tau - x \sin\tau$$

where superscript R means rotated. All subsequent use of x and y in the curve fit section is assumed to mean rotated, and the superscript will be dropped.

4.8.4 Formation and Solution of Normals

Since the polynomial is linear, solution may be made directly by

$$\begin{aligned} x &= a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t_4 + a_5 t^5 \\ y &= b_0 + b_1 t + b_2 t^2 + b_3 t^3 + b_4 t_4 + b_5 t^5 \end{aligned} \tag{41}$$

where $t = N_i - N_c$, and x and y are rotated. Since the normal matrix for the two equations is the same, two absolute columns are set up and both are solved in one operation. The coefficients of the two polynomials are the result.

4.8.5 Residuals and Mean Error

Using the computed coefficients, the polynomial is evaluated at each point by eq.(41), and the residuals in x and y are computed by

$$v_x = x^c - x$$

$$v_y = y^c - y$$

where x and y are the original rotated coordinates, and x^c and y^c are computed.

The mean errors for a fifth-order polynomial are:

$$s_x{}^C = \left(\frac{\sum v_x{}^2}{N-6}\right)^{\frac{1}{2}}$$

$$s_y{}^C = \left(\frac{\sum v_y{}^2}{N-6}\right)^{\frac{1}{2}}$$

$$s^C = \left(\frac{\sum v_x{}^2+\sum v_y{}^2}{2(N-6)}\right)^{\frac{1}{2}} \tag{42}$$

where superscript C means curve fit. These give a measure of the trail's lack of adherence to a smooth curve, the x in the direction of motion, and the y perpendicular to it.

4.8.6 Residual Check

From experience, the criterion was established that any residual greater than 20μm must represent an erroneous point. If any such points are present, they are eliminated and the solution is repeated. New residuals and new mean errors are computed.

4.8.7 Graphical Analysis of Residuals: Histogram

In a procedure similar to that for differences after matching, the residuals in x and y after curve fit are arranged in histograms and plotted. A normal curve [eq.(39)] is superimposed on each histogram. As in matching, a significant departure from the normal would show a need for further investigation.

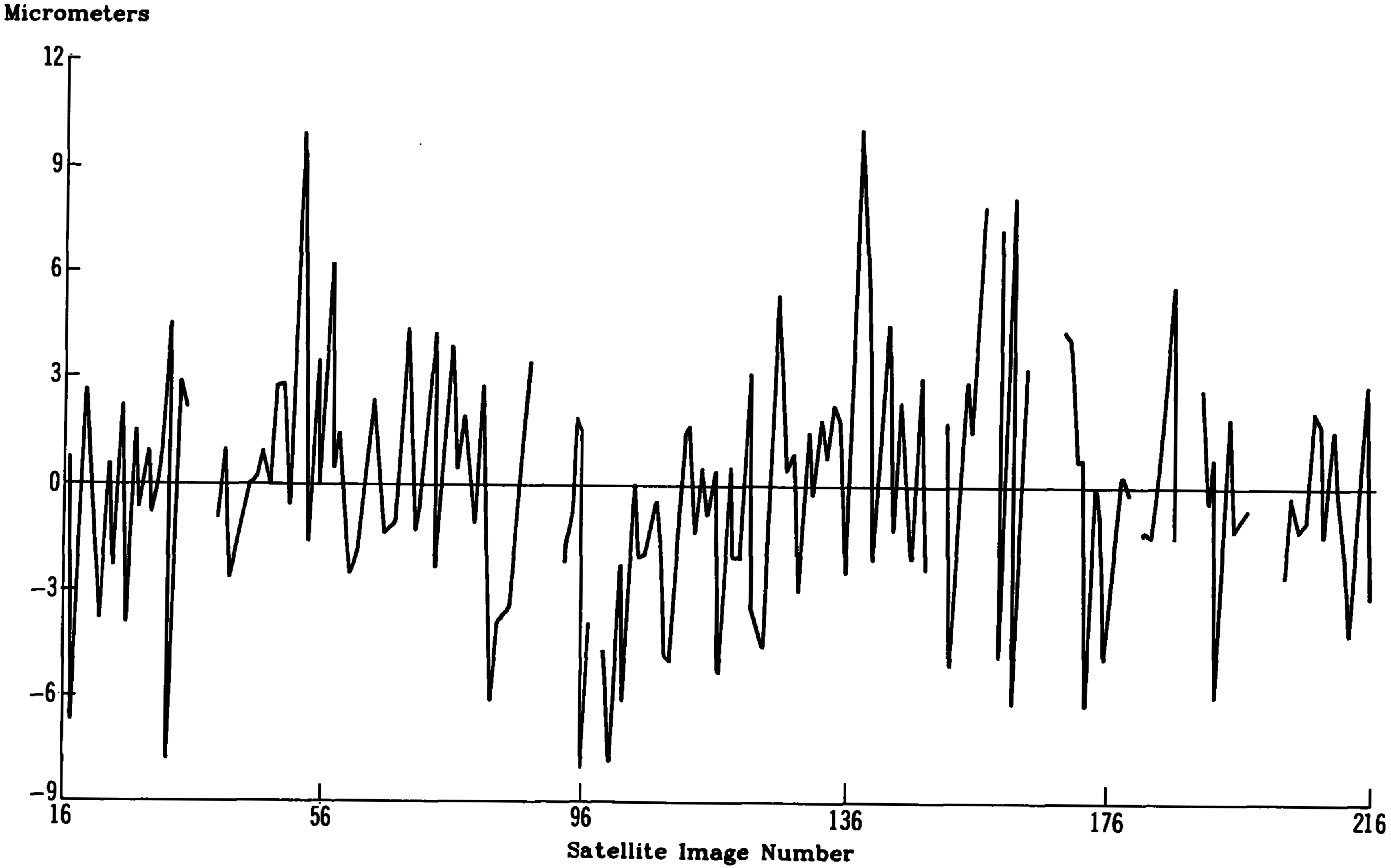
4.8.8 Graphical Analysis of Residuals: Frequency Plot

The mean errors and histograms are averages over the whole curve. It is possible that the mean error or the distribution might differ from one part of the curve to another; such differences signal plate distortion or systematic error of some kind. For this reason, the residuals in x and in y are plotted against their position along the satellite's track length as represented by the image number. The resulting chart resembles a frequency plot, so it has been given that name. Figure 6 shows a section of a frequency plot. Visual inspection can be used to determine if the pattern of residuals is well-behaved.

4.9 Output

4.9.1 Files of Data

The output of the Plate Data Reduction program consists of six files of data. The first two are the raw data sets A and B with images numbered (described in sec. 4.5.6). The next file is the output of matching, and the fourth file is the output of satellite transformation. These comprise the plate measurement data that will be used in all future computations. The final two files are the differences after matching and the residuals after curve fit. These are stored on magnetic tape for possible more detailed analysis in the future.


Micrometers
12
9
6
3
0
−3
−6
−9
16
56
96
136
176
216
Satellite Image Number

4.9.2 Weights

The star position records in file 3 and the satellite position records in file 4 also contain three terms of a weight matrix: the weights for the $x-$ and $y-$coordinates and the correlation term. In the past, experimentation was done on individual weighting of points, based on factors such as distance from the plate center and mean error of measurement (if the point was measured more than once). It was decided to eliminate individual weighting and instead to assign a weight to the plate as a whole. So all image records carry the individual weight matrix of 1, 1, 0.

The plate weight is computed and printed after the curve fit routine. It is based on the mean error of the star images. The mean error of the curve fit is assumed equivalent to the mean error of the satellite images. The mean error of unit weight of the Single Camera Orientation, for which the plate weight is being computed, is taken as 2.5μm.

$$s^2_{\text{star}} = s^2_{\text{star measurement}} + s^2_{\text{emulsion}} + s^2_{\text{scintillation}}$$

$$s^2_{\text{curve fit}} = s^2_{\text{satellite measurement}} + s^2_{\text{emulsion}} + s^2_{\text{scintillation}}$$

Therefore,

$$\begin{aligned} s^2_{\text{star}} &= s^2_{\text{star meas.}} + s^2_{\text{curve fit}} - s^2_{\text{sat.meas.}} \\ &= (s^M)^2 + (s^C)^2 - (s^T)^2 \end{aligned} \qquad (43)$$

where s^M, s^T, and s^C are from eq.(38), (40), and (42) respectively. Plate weight P_l is computed by

$$P_l = \frac{(2.5)^2}{s^2_{\text{star}}}. \qquad (44)$$

5. STAR IDENTIFICATION

5.1 Introduction

The final step in preparing the input for the Single Camera Orientation program (Slama 1972) is to associate each measured star image with the celestial coordinates of the star and with the time of exposure. This is the purpose of the last of the three major computer programs, Star Identification and Updating, program 380 on the CDC 6600.

The input consists of:

a. tolerances for star lookup, usually 3' of arc in right ascension and in declination;
b. weights to be assigned to the stars' celestial positions according to the source of catalog information;
c. approximate camera orientation parameters (from Preliminary Reduction);
d. Independent Day Number parameters for all exposure times (sec. 1.3.4);
e. reduced star measurements (file 3 of output of Plate Data Reduction).

There is also standard input consisting of the 20,291 stars selected from the SAO catalog, as discussed in sec. 2.2.1, and a list of second-order day numbers J and J' taken from the American Ephemeris and Nautical Almanac. These are maintained on magnetic tape. At the beginning of every run of the Star Identification program these are stored on the computer disk for access during processing.

The method to be followed is to compute the approximate celestial coordinates for each star image, then to search the catalog for an acceptable match. The time of exposure is located by matching time codes.

5.2 Initial Steps of Processing

The direction cosine matrix is computed from the approximate camera orientation by eq.(22). The Independent Day Number parameters are read in and stored in array DAYNUM. For each of up to 90 exposure times there are eight parameters in storage: the local sidereal time of exposure, and the C, H, f, g, h, i, and τ parameters. The corresponding time codes are stored in XTRL. DAYNUM and XTRL are sorted on increasing time code.

The star image measurements are read in and stored in array DATA. The seven pieces of information for each image include x and y measurements, weights in x and y and their correlation, the image number, and the time code. DATA is sorted on increasing image number.

5.3 Approximate Celestial Coordinates

5.3.1 Coordinates at Time of Exposure

Using the plate measurements and the direction cosine matrix, approximate standard coordinates are computed for each image, using eq.(20) and (23). These in turn are used to compute the azimuth and zenith distance by eq.(25). Since this zenith distance is computed from a plate position, it is necessary to remove the effects of refraction (neglecting diurnal aberration) to get a better approximation of the star's position by

$$z' = z + R$$

where z is the distance as computed, z' is after refraction effects are removed, and R is as defined in eq.(13).

The hour angle and declination are then computed by eq.(26). To compute the right ascension it is necessary to know the local sidereal time. The time code for each image is matched with the time codes in XTRL, and the appropriate set of parameters located in DAYNUM.

5.3.2 Coordinates as of 1950.0

The local sidereal time is stored in the seventh element of DATA for the image, replacing the time code which is no longer needed. The star position is backdated to the nearest Besselian year, the reverse of the process in eq.(11), and neglecting proper motion and the J and J' terms.

$$\alpha' = \alpha - \left[f + g \sin(\alpha + G) \tan\delta + \frac{h \sin(\alpha + H)}{\cos\delta} \right]$$

$$\delta' = \delta - [g \cos(\alpha + G) + h \cos(\alpha + H) \sin\delta + i \cos\delta]$$

In the early days of satellite triangulation, prior to the publication of the SAO catalog, the Boss and N30 catalogs were used. With these, the method of annual and secular variation was used for updating and also for backdating the approximate positions. When the SAO catalog and the Scott-Hughes (1964) method of updating came into use, it was not necessary to change the backdating procedure because this usage is an approximation. Proper motion is neglected. In radians:

$$\begin{aligned} \alpha_{50} &= \alpha' - \frac{\Delta T}{100} I - \left(\frac{\Delta T}{100}\right)^2 II \\ \delta_{50} &= \delta' - \frac{\Delta T}{100} I' - \left(\frac{\Delta T}{100}\right)^2 II' \end{aligned} \qquad (45)$$

where α_{50}, δ_{50} are the right ascension and declination for 1950, and

$$I = .02234945 + .0097169024 \sin\alpha \tan\delta$$

$$I' = .0097169024 \cos\alpha$$

$$II = .000006763151 + (.0048584512\, I \cos\alpha - .0000020604581 \sin\alpha)\tan\delta + .0048584512\, I' \sin\alpha / \cos^2\delta$$

$$II' = -.0000020604581 \cos\alpha - .0048584512\, I \sin\alpha.$$

5.3.3 Storage of 1950 Coordinates

Each series of (usually) five images in a trail will have approximately the same star coordinates. A mean is taken of these to arrive at a single approximation for each star measured. The resulting positions are stored in array COORD. Also in COORD are the mean declination for the time of observation, the star number (location code plus trail number), the number of images in the trail, and the position in array DATA of the first image of the star. After all stars have been entered in COORD, the array is sorted on increasing right ascension.

5.4 Star Lookup

5.4.1 Catalog Information

The catalog is stored on the disk in 298 groups of 68 stars each, plus a partial 299th group, arranged in order of increasing right ascension. There are six pieces of information for each star: 1950 right ascension and declination, proper motion in each, catalog number, and source catalog. The last two items require a brief explanation. The catalog number is not the one assigned by SAO, but one used for internal convenience only. The source catalog is the origin of the information as listed in the SAO catalog, which was compiled from previous sources. The FK4 catalog is known to be more accurate than other sources; therefore, stars from this catalog are to be given more weight in the Single Camera Orientation procedure. An array RTSV in core contains the maximum right ascension of the data blocks on the disk.

5.4.2 Principles of Catalog Search

a. The range of possible values of right ascension is computed as the approximate right ascension minus and plus a tolerance. Within this range, any star within the declination tolerance is assumed a match.

b. Matching with the updated rather than the 1950 position is the final criterion.

c. Binaries have not been completely removed from the catalog, so it is possible that a measured star could pass all tolerance tests with the wrong star of a pair. For this reason a test must be made that there are not a pair of stars meeting the criteria; if there are, the measured star is not identified, but remains in the solution as an unknown.

d. The stars from the catalog actually searched for each identification are limited to a maximum of 68: either a single block from the disk, or a composite of two contiguous half blocks. Experience has shown that in practically all cases the range of right ascensions falls within such a grouping.

5.4.3 Location of Right Ascension Range in Catalog

After a star's range has been computed, the program logic, by comparison with array RTSV and with the section of catalog currently in core, determines whether the range is contained within the current section, within a single block on the disk, within two contiguous half blocks on the disk, or falls under one of three special cases concerned with the upper and lower ends of the catalog. The appropriate section of the catalog is then brought into array STARS.

5.4.4 Determination of Match

For each catalog star within the right ascension range, the declination is tested. If it falls within the declination range, the catalog star is updated to the initial time of observation for the trail, and the declinations are again compared. The position of the star for both 1950 and time of observation is stored in array

ST, together with the star weight matrix assigned according to the source of the catalog information: 2.95, 2.95, 0 is used for FK4 stars, and 1.66, 1.66, 0 for all others. (These are based on a mean error of .3" for the FK4 stars and .4" for others, with a mean error of unit weight of 2.5μm.) There are two columns in ST, so that if two matches are found they are both stored, the one with the updated declination closest to the original approximation in the first column.

After all stars within the range have been checked, a decision is made whether the star is identified, unidentified, or binary. The last is the case if there are two stars in array ST whose updated declinations are less than 20" apart. If no matches were found, it is unidentified. If one match was found, or two matches were found with more than 20" difference in declination, the star in the first column of ST is taken as the identity.

For an identified star, the updated coordinates in ST are transferred to COORD, replacing the approximate coordinates. For unidentified and binary stars, the star number in COORD is made negative as an indicator.

5.4.5 Unidentified Stars

A star will not identify for one of several reasons:

a. an error was made in choosing for measuring an image that did not appear on the star chart, *i.e.* is not in the selected catalog,

b. the renumbering procedure in section 4.6.4 assigned the wrong trail number to an image, so that it was later associated with the wrong exposure time, or

c. the approximate camera orientation parameters are incorrect. In this case many or all stars will fail to identify so the error will be obvious.

5.5 Output

5.5.1 Combined Trails

The stars in COORD are rearranged so that stars with the same catalog number, if any, are together, and all unidentified stars are grouped at the end. As discussed in section 1.3.3, several trails of the same star may be measured. The differing backdating parameters applied to the coordinates of these images in different areas of the plate result in approximately the same 1950.0 star coordinates; the same star will be identified for each of the trails.

Since the celestial position of this star represents only two pieces of information no matter how many times it is used, it is necessary to combine all the trails of images for the input to Single Camera Orientation. To signal that this has been done, the star number is changed to an arbitrary number, sequentially from 1 for the first combined star on the plate.

5.5.2 Output Star Record

The updated star position is output from COORD to magnetic tape. In the case of a combined star, the star number is changed as explained above, and the number of images is now the total for all trails; for an unidentified star the only quantities output are the number of images and the star number. The format used is:

rt. asc.	dec.	star weight		star no.		no. of		catalog
		matrix				images		no.
E14.7,	E14.7,	3E10.3,	2X,	I6,	2X,	I3,	1X,	A6

The star record is immediately followed by its associated images.

5.5.3 Output Image Records

The associated image data for the star is located in DATA by means of the initial location stored in COORD. If the star was combined, as many trails as necessary are located. Each image is output to magnetic tape in format:

x	y	plate weight		image	star	LST
		matrix		no.	no.	
F12.10,	F12.10,	3E10.3,	2X,	I6,	I6,	F11.8

6. SATELLITE DATA

The contents of file 4 of the output of Plate Data Reduction are now ready to be used as input to the Satellite Image Reduction program (Hanson 1973) without further treatment. The format of each record is:

x	y	plate weight matrix	class	image no.
E14.7,	E14.7,	3E10.3,	I2,	I6

Also input to the Satellite Image Reduction program are the camera orientation parameters resulting from Single Camera Orientation, plus exposure time for the initial image, and time interval between exposures for the plate. The exposure time for each satellite image after the initial image will be computed in the program:

$$T_j = T_i + (N_j - N_i)\,\Delta \qquad (46)$$

where i refers to the initial image, N is image number, and Δ the time interval. For the BC-4 camera, Δ is assumed constant to 10 microseconds. See appendix B for modifications necessary for other cameras.

ACKNOWLEDGEMENTS

The Satellite Triangulation project is under the scientific direction of Dr. Hellmut H. Schmid. All the plate measurements and data handling have been done by the Data Reduction Branch of the Satellite and Marine Applications Division, National Geodetic Survey, headed by Raymond J. Puhl. Portions of the program coding were done by Wallace H. Blackwell, James Lisle, and Clarence G. Beale. The author is indebted to Robert H. Hanson and Chester C Slama for much helpful advice.

REFERENCES

Bossler, John D., *The SAO Catalog, Its Qualitative and Quantitative Value to the C&GS Satellite Triangulation Program,* Evironmental Science Services Administration, Rockville, Md., Aug. 1966, 52 pp.

Hanson, Robert H. (Geodetic Research and Development Laboratory, National Ocean Survey, National Oceanic and Atmospheric Administration, Rockville, Md.), "Satellite Image Reduction," 1973 (in preparation).

Schmid, Erwin (Geodetic Research and Development Laboratory, National Ocean Survey, National Oceanic and Atmospheric Administration, Rockville, Md.), 1965a (private communication).

Schmid, Erwin, "Program to Compute and Plot Histogram and Gaussian Curve," 1965b (unpublished manuscript).

Schmid, Erwin, *Cholesky Factorization and Matrix Inversion,* National Oceanic and Atmospheric Administration, Rockville, Md., March 1973, 13pp.

Schmid, Hellmut H., "Triangulation Mit Satelliten" (Triangulation Using Satellites), *J.E.K.–Handbuch der Vermessungskunde,* Vol. III a/2, Metzler Verlag, Stuttgart, Germany, 1972, pp. 2081–2233.

Scott, F.P., And Hughes, J.A., "Computation of Apparent Places for the Southern Reference Star Program," *Astronomical Journal,* Vol. 69, No. 5, New York, N.Y., June 1964, pp. 368–371.

Slama, Chester C, *A Mathematical Model for the Simulation of a Photogrammetric Camera Using Stellar Control,* National Oceanic and Atmospheric Administration, Rockville, Md., Dec. 1972, 138 pp.

Smithsonian Institution, *Smithsonian Astrophysical Observatory Star Catalog,* 4 vol., Washington, D.C., 1966, unpaged.

U.S. Government Printing Office, *American Ephemeris and Nautical Almanac,* Washington, D.C., yearly.

APPENDIX A. PROVISIONS FOR REPROCESSING DATA

At the time of the reformulation in 1968, all plates already processed were rerun under the new formulation. A data management decision had previously been made that the raw data directly from the comparator was not to be retained, but the first-step results of numbered images were to be accepted as a sufficient base. So, for rerun purposes, a program was written that is parallel to Plate Data Reduction, but takes into account the differences in input. This program is referred to as 382.

Even though the rerun has now been completed, there are circumstances under which a plate currently being processed might be rerun shortly after its initial passage through Plate Data Reduction. The most common reason is a badly measured point (or points) which show up with high residuals after matching or after transformation of satellite images. It is desirable to rerun with these points removed to obtain results undistorted by their presence. Another possible occurence is that of a plate rotated by an amount significantly different from 180° between the A and B sets. Unless this is known in advance, the input tolerance for the pre-matching check (sec. 4.6.5) will be the standard 100μm and many images will be rejected on this basis. When this has been noted in the first run, an immediate rerun takes place with a higher tolerance. Because the use of a standard procedure for rerun is efficient and convenient, all reruns take place on 382 even though the original input to 383 may still be available. The input to 382 consists of the first two files of output of 383.

The special characteristics of 382 have been determined by its double purpose of rerunning both old and new plates. For the old plates it contains various options to allow needed flexibility in the input data, because the plates were measured in various ways through the years of development. For the new plates, the default condition on all the options is such that the numbered raw data output from 383 goes directly in.

The most substantial difference between 382 and 383 is that 382 does not contain the star-numbering routine. Since this procedure is quite insensitive to errors in input assumptions, being concerned only with relative position, it is assumed that once the star images are numbered there is no need for a repetition. The satellite numbering routine, on the other hand, is sensitive to various errors and is a frequent reason for rerunning. So even though the satellite images have numbers in the input to 382, these are ignored and the procedure is repeated in the program.

To carry out its function of rerunning current plates with problem conditions, 382 contains several options for manipulating data prior to the various processing steps.

a. Up to 100 images of any type may be rejected at the time of input. This permits the removal of trouble-making images without the necessity for recreating the data file containing the input.

b. If there was a large plate shift between subsets, the processing of the plate could have been completed by using a large tolerance for the pre-patching test. But this masks the real purpose of the test, which is to test for irregularities in the measurement of the drill holes. To get a better picture of the patching operation, the large shift computed during the first patching run may be applied to the drill hole positions *prior* to patching in the rerun. The standard tolerance may then be used, and the patching computation can reveal small differences in the drill hole measurements, as it is intended to do.

c. The same principle is true for the matching operation; in fact, differences between the A and B sets are more common and more important than those between subsets. There is also an option to apply a shift or rotation to the B set prior to matching.

d. Another common reason for rerunning, erroneous satellite numbering, is caused by problems in the Pre set. A rerun of Preliminary Reduction may be necessary, but no special options are needed in 382.

APPENDIX B. CAMERAS OTHER THAN BC-4

The BC-4 ballistic camera, used by the National Geodetic Survey for the Satellite Triangulation project, has a very accurate shutter mechanism; exposure times can be controlled to 10 microseconds. As discussed in section 6, the procedure for associating each satellite image with its exposure time depends on this accuracy.

Certain stations in the Satellite Triangulation system have been operated by other groups cooperating with NGS, but using different kinds of equipment. Since NGS has the responsibility for measuring and processing all plates, a different satellite procedure had to be developed for plates exposed by cooperating agencies. This procedure is embodied in program 697, Plate Data Reduction for Non-Uniform Times.

The brush tape time record tells the times at which the shutter was open, so each satellite image can be associated with its time. The problem is how this can be done efficiently and in such a way that the results can be merged with those plates exposed during the same event in BC-4 cameras.

A different system of numbering is used, in which the image number represents actual time rather than a number of increments of time. A base time is selected, which must be the same as or earlier than the initial image times of all plates in the event, and must be in the exposure time series of the plate(s) exposed in BC-4 cameras, so it can be used as time 1 in this series. For the nonuniform plates the image numbers are computed from:

$$N_j = (T_j - T_b)\,10^5$$

where T_j and T_b are the exposure time and base time in seconds.

The star image numbering is also done in a preliminary run. Then the data are processed in program 697, which is parallel to 383 in most ways except for the absence of image numbering. Because of the different form of satellite image numbers, there are differences in storage methods, and scale factors are applied to image numbers in curve fit and the frequency plot for more convenient handling of the large numbers.

When satellite coordinate records are input to Satellite Image Reduction (Hanson 1973), a signal indicates whether or not the camera had uniform timing. For uniform plates, eq.(46) is used to compute exposure time. In the case of nonuniform timing, the exposure time is computed as:

$$T_j = T_i + (N_j - N_i)\,10^{-5}$$

where i refers to the initial image on the plate.

APPENDIX C. PROGRAM LISTINGS

```
      PROGRAM PRELIM (TAPE2,TAPE4,OUTPUT)
      DIMENSION A(6),IPUN(3),AN(6),DATE(2),OPER(3),STSA(2),STLO(2),
     1 IST(6),STWE(2),REC(20),RL(34),XF(8),YF(8),DSX(20),DSY(20),
     2 IDSD(20),IDPT(20),X(10),Y(10),NXY(10),NUM(10),ITRL(90),R(21),
     3 STL(9),STR(90),GCPR(90),HCPR(90),FFR(90),GSMR(90),HSMR(90),
     4 SMIR(90),TAU(90),ITRA(10),Q(3),DC(3,3),QQ(3),ALS(10),DLS(10),
     5 PSI(10),ADA(10),CATNA(10),F(8,12),S(20,9),VS(10,2)
      COMMON /CDBUF/ LENGT,NEXT,JFIRST,INBUF,BUFF(1024),JS,IP
      COMMON /ODBUF/ LENG,NEX,IFIR,IBUF,BUF(1024),ENDFLQ,IQ
      DATA (RDG=.17453292519943E-1),(RMIN=.29088820866572E-3),
     1 (RSEC=.48481368110953E-5),(RHR=.26179938779915),
     2 (RTMIN=.43633231299858E-2),(RTSEC=.7272205216643E-4),
     3 (RDTOGR=63.661977),(AP0=.29137566E-3),(AP1=.3227865E-6),
     4 (AP2=.10225E-8),(PCON=6370000.),(BETA=.3665E-2),(P0=76C.),
     5 (T0=0.),(IDUM1=13),(IDUM2=5),(IDUM3=43),(IDUM5=71),(IDUM0=0),
     6 (ZERO=0.),(BLANK=1H )
      DATA (IP=2401200535000000000B),(IQ=2401200537000000000B),
     1 (IEF=10000000000000B)
      IZYX = 0
      CALL PATCH
      IF(IZYX.EQ.0) GO TO 100
      PRINT 899,(BUFF(I),I=1,1024)
      IF((JS.AND.IEF) .NE. 0) GO TO 100
      CALL LTRIO (IP,4B,XX,XX,JS)
  100 IZYX = 1
      IF(ENDFLQ.EQ.0.) CALL LTRIO (IQ,115B,XX,XX,KS)
      ENDFLQ = 1.
      LENGT □ LENG = 0
      NEXT = JFIRST = 1
      NEX = IFIR = 1
      CALL DBUF(0)
      CALL DBUF(1)
      DECODE (80,1,BUFF(INBUF)) ISTEP,(A(I),I=1,6)
      PRINT 801,ISTEP,(A(I),I=1,6)
      CALL OBUF(1)
      ENCODE (80,20,BUF(IBUF)) ISTEP,(A(I),I=1,6)
C
C  READ AND PRINT GENERAL INFORMATION
      CALL DBUF(1)
      DECODE (80,2,BUFF(INBUF)) EV,PL,COMM,IODR,(IPUN(I),I=1,3),ISTAT
     1 ,IPRE
      IF(IODR.LT.1) IODR □ 0
      DO 102 I=1,3
      IF(IPUN(I).LT.1) IPUN(I) = 0
  102 CONTINUE
      CALL DBUF(1)
      DECODE (80,3,BUFF(INBUF)) ISTA1,(AN(I),I=1,3),ISIGS,PHILH,PHILM,
     1 PHILS,XLAMLH,XLAMLM,XLAMLS,EL1
      CALL DBUF(1)
      DECODE (80,3,BUFF(INBUF)) ISTA2,(AN(I),I=4,6),ISIGR,PHIRH,PHIRM,
     1 PHIRS,XLAMRH,XLAMRM,XLAMRS,EL2
      CALL DBUF(1)
      DECODE (80,4,BUFF(INBUF)) CAM,UNIT,DATE,OPER
      CALL DBUF(1)
      DECODE (80,5,BUFF(INBJF)) TEM,PRES,XPC,YPC
      IPAGE=1
      PRINT 802,IPAGE
      PRINT 803,EV,PL
      PRINT 804,COMM
      PRINT 805,IODR,IPUN
      PRINT 806,CAM,UNIT,DATE,OPER
      PRINT 807,TEM,PRES,XPC,YPC
      IF(ISIGS.LT.1)ISIGS=0
      PRINT 808
      PRINT 810,ISTA1,(AN(I),I=1,3),ISIGS,PHILH,PHILM,PHILS,XLAMLH,
     1  XLAMLM,XLAMLS,EL1
      IF(ISTA2.NE.ISTAT) PRINT 901,ISTAT
      IF(ISIGR.LT.1)ISIGR=0
      PRINT 809
      PRINT 810,ISTA2,(AN(I),I=4,6),ISIGR,PHIRH,PHIRM,PHIRS,XLAMRH,
     1  XLAMRM,XLAMRS,EL2
      IF(IODR.EQ.1) GO TO 105
C
C  READ AND PRINT DATA RECORD (OPTIONAL)
      CALL DBUF(1)
      DECODE (80,6,BUFF(INBUF)) STSA,STLO,STLA,STSR
      CALL DBUF(1)
      DECODE (80,7,BUFF(INBUF)) (IST(I),I=1,6),EMT,STWE
      CALL DBUF(1)
      DECODE (80,8,BUFF(INBUF)) (REC(I),I=1,11)
      CALL DBUF(1)
      DECODE (80,9,BUFF(INBUF)) (REC(I),I=12,17)
      CALL DBUF(1)
      DECODE (80,10,BUFF(INBUF)) (IST(I),I=7,12),(REC(I),I=18,20)
      IPAGE=IPAGE+1
      PRINT 811,IPAGE
      PRINT 812,EV,PL,STSA,DATE
      PRINT 813,ISTA1,(AN(I),I=1,3),STLO,PHILH,PHILM,PHILS,XLAMLH,
     1  XLAMLM,XLAMLS,EL1
      PRINT 814,UNIT,CAM,STLA,(IST(I),I=1,6),STSR,EMT,PRES,STWE
      PRINT 815,(REC(I),I=4,7),(REC(I),I=1,3),(REC(I),I=8,11),(REC(I),
     1  I=18,20)
      PRINT 816,REC(12),REC(14),REC(16),REC(13),REC(15),REC(17)
      PRINT 817,(IST(I),I=7,12),XPC,YPC
      PRINT 818
  103 CALL DBUF(1)
      DECODE (80,11,BUFF(INBUF)) IRMT
      PRINT 11
      IF(IRMT.EQ.0) GO TO 103
C
C  READ IN COMPARATOR PARAMETERS
  105 CALL DBUF(1)
      DECODE (80,12,BUFF(INBUF)) CRI
      CALL DBUF(1)
      DECODE (80,13,BUFF(INBUF)) ICOM,(RL(I),I=1,4)
      DO 107 I=5,30,5
      J = I+4
      CALL DBUF(1)
      DECODE (80,12,BUFF(INBUF)) (RL(K),K=I,J)
  107 CONTINUE
      IPAGE=IPAGE+1
      PRINT 819,IPAGE
      PRINT 803,EV,PL
      PRINT 820,ICOM
      PRINT 821
      PRINT 822
      NCT=12
      SINA=RL(17)
C
C  READ IN MEASURED PRE SET
      J=NOPT=M=ISR=IDH=IND=0
  110 CALL DBUF(1)
      DECODE (80,14,BUFF(INBUF)) NPLT,ITYP,NPT,G,H,ITEST
      IF(ITYP.GT.7) GO TO 115
      IF(NCT.LT.64) GO TO 112
      IPAGE = IPAGE+1
      PRINT 819,IPAGE
      PRINT 823
      PRINT 803,EV,PL
      PRINT 822
      NCT □ 9
  112 IG = G*1.E+6 + .1
      IH = H*1.E+6 + .1
      PRINT 824,ITYP,NPT,IG,IH
      NCT=NCT+1
C
C  MEAN MEASUREMENTS OF SAME POINT
  115 IF(NOPT.NE.0) GO TO 118
  117 NOPT = NPT
      ID = ITYP
      N=SUMX=SUMY=0.
  118 IF(NPT.NE.NOPT.OR.ID.NE.ITYP) GO TO 125
      N = N+1
      IF(N.LE.25) GO TO 120
      PRINT 902,NPT
      NCT=NCT+3
      N=N-1
      GO TO 122
  120 SUMX = SUMX + G
      SUMY □ SUMY + H
  122 IF(ITEST.EQ.0) GO TO 110
  125 TN = N
      X2 □ SUMX/TN
      Y2 □ SUMY/TN
C
C  APPLY COMPARATOR REDUCTION
  130 X6 = X2*RL(14) + Y2*SINA*RL(15)
      Y6 □ Y2*RL(15)
      IF(ID.EQ.3) GO TO 131
      IF(M.EQ.8) GO TO 142
      PRINT 903
      CALL LTRIO (IP,4B,XX,XX,KS)
      GO TO 100
C
C  STORE DRILL HOLES FOR COMPUTATION
  131 M □ M+1
      IF(NOPT/100.EQ.M) GO TO 132
      PRINT 903
      GO 10 117
  132 XF(M) = X6
      YF(M) = Y6
      IF(M-8) 117,135,131
C
C  COMPUTE CENTER OF DRILL HOLES
  135 CALL PLTCEN(XF,YF,XC,YC)
      D13 □ SQRT((XF(3)-XF(1))**2 + (YF(3)-YF(1))**2)
      D24 = SQRT((XF(4)-XF(2))**2 + (YF(4)-YF(2))**2)
C
C  COMPUTE ROTATION TO FIDUCIAL SYSTEM, AND ORIENTATION OF COORDINATES
      DELX □ XF(3) - XF(1)
      DELY □ YF(3) - YF(1)
      CALL ANGLE(DELY,DELX,TAUL)
      SNTL=SIN(TAUL)
      CSTL=COS(TAUL)
      YCON=1.
      F2 = (YF(2)-YC)*CSTL - (XF(2)-XC)*SNTL
      F4 = (YF(4)-YC)*CSTL - (XF(4)-XC)*SNTL
      IF(F4.LT.F2) YCON □ -1.
C
C  STORE DRILL HOLES AND SATELLITES FOR OUTPUT
      II = 0
  138 II = II+1
      X6 = XF(II)
      Y6 = YF(II)
      NOPT=II*100
      ID=3
  142 X11 = X6 - XC
      Y11 = (Y6 - YC)*YCON
      IF(ID.EQ.2) GO TO 145
      IDH □ IDH+1
      IF(IDH.LE.20) GO TO 143
      PRINT 905
      GO TO 150
  143 DSX(IDH)=X11
      DSY(IDH)=Y11
      IDSD(IDH)=ID
      IDPT(IDH)=NOPT
      IF(II-8) 138,150,150
C
C  STORE STARS FOR OUTPUT
  145 ISR □ ISR+1
      IF(ISR.LE.10) GO TO 147
      PRINT 904
      GO TO 150
  147 X(ISR)=X11
      Y(ISR)=Y11
      NXY(ISR)=NOPT
      NUM(ISR) = NOPT/1000
  150 IF(ITEST.EQ.1) GC TO 160
C
C  PROCESS SATELLITES
      IF(ITYP.LE.7) GO TO 117
  151 IF(NCT.LT.64) GO TO 152
      IPAGE = IPAGE+1
      PRINT 819,IPAGE
      PRINT 823
      PRINT 803,EV,PL
      PRINT 822
      NCT = 9
  152 IG = G*1.E+6 + .1
      IH = H*1.E+6 + .1
      PRINT 824,ITYP,NPT,IG,IH
      NCT=NCT+1
      IF(IND.EQ.0) GO TO 155
      IND = 0
      GO TO 130
  155 X2 = G
      Y2 = H
      NOPT = NPT
      ID = ITYP
      N = 1
      CALL DBUF(1)
      DECODE (80,14,BUFF(INBUF)) NPLT,ITYP,NPT,G,H,ITEST
      IF(ITYP.GE.7) GO TO 130
      IF(ITEST.EQ.1) GO TO 130
      IND = 1
      GO TO 151
C
C  OUTPUT DRILL HOLES, SATELLITES AND STARS (OPTIONAL)
  160 IPAGE = IPAGE+1
      PRINT 819,IPAGE
      PRINT 823
      PRINT 803,EV,PL
      PRINT 825
      PRINT 826,D13,D24,XC,YC
      PRINT 827
      CALL OBUF(1)
      ENCODE (80,50,BUF(IBUF)) IDUM1
      IEND = 0
      DO 170 I=1,IDH
      IF(IPUN(1).EQ.1) GO TO 168
      IF(I.EQ.IDH) IEND = 1
      CALL OBUF(1)
      ENCODE (80,21,BUF(IBUF)) DSX(I),DSY(I),IDSD(I),IDPT(I),PL,IEND
  168 PRINT 828,DSX(I),DSY(I),IDSD(I),IDPT(I),PL
  170 CONTINUE
      CALL OBUF(1)
      ENCODE (80,50,BUF(IBUF)) IDUM1
      DO 175 I=1,ISR
      ID = 2
      IF(IPUN(2).EQ.1) GO TO 173
      CALL OBUF(1)
      ENCODE (80,21,BUF(IBUF)) X(I),Y(I),ID,NXY(I),PL
  173 PRINT 828,X(I),Y(I),ID,NXY(I),PL
  175 CONTINUE
C
C  READ IN UPDATING PARAMETERS
      IPAGE=IPAGE+1
      PRINT 829,IPAGE
      PRINT 803,EV,PL
      PRINT 821
      PRINT 830
      NCT=17
      IIH □ 0
      J = IS = 1
      CALL DBUF(1)
      DECODE (80,15,BUFF(INBUF)) ITRL(J),PLT,IYEAR,MONTH,JDAY
      IF(PL.EQ.PLT) GO TO 185
      PRINT 910,PL,PLT
      CALL LTRIO (IP,4B,XX,XX,JS)
      GO TO 100
  185 CALL DBUF(1)
      DECODE (80,12,BUFF(INBUF)) (R(I),I=1,3)
      CALL DBUF(1)
      CALL DBUF(1)
      DECODE (80,12,BUFF(INBUF)) (R(I),I=4,6)
      CALL DBUF(1)
      DO 190 I=7,19,3
      I2 □ I+2
      CALL DBUF(1)
      DECODE (80,12,BUFF(INBUF)) (R(K),K=I,I2)
  190 CONTINUE
      IF(NCT.LE.53) GO TO 192
      IPAGE = IPAGE+1
      PRINT 829,IPAGE
      PRINT 823
      PRINT 803,EV,PL
      NCT □ 7
  192 PRINT 831,ITRL(J),PLT,IYEAR,MONTH,JDAY
      PRINT 832,(R(I),I=1,21)
      NCT=NCT+9
      PHI = R(1)*RDG + R(2)*RMIN + R(3)*RSEC
      STR(J)  □ R(4)*RHR + R(5)*RTMIN + R(6)*RTSEC
      IT □ ITRL(J)
      STL(IT) = STR(J)
      GCPR(J) = R(7)*RHR + R(8)*RTMIN + R(9)*RTSEC
      HCPR(J) =R(10)*RHR +R(11)*RTMIN +R(12)*RTSEC
      FFR(J)  □ R(13)*RTSEC
      GSMR(J) □ R(14)*RSEC
      HSMR(J) □ R(15)*RSEC
      SMIR(J) = R(16)*RSEC
      TAU(J)  □ R(17)
      J = J+1
C
C  READ IN STAR POSITION
  195 CALL DBUF(1)
      DECODE (80,16,BUFF(INBUF)) RA,PMRA,DEC,PMDC,CATNO,MTYP,MPT,
     1  ITRL(J),PLT,MTEST
      IF(MTYP.EQ.0) GO TO 185
      JJ = J-1
      IF(IIH.EQ.1) GO TO 200
      IF(NCT.LE.52) GO TO 197
      IPAGE = IPAGE+1
      PRINT 829,IPAGE
      PRINT 823
      PRINT 803,EV,PL
      NCT = 7
  197 PRINT 833
      IIH=1
      NCT = NCT+5
```

```
C
C  MATCH STAR POSITION WITH STAR MEASUREMENTS
  200 DO 205 L=IS,ISR
      IF(NUM(L).EQ.MPT) GO TO 210
  205 CONTINUE
      IF(MTEST.EQ.0) GO TO 195
      GO TO 260
  210 IF(L.EQ.IS) GO TO 212
      TEMP = X(L)
      X(L) = X(IS)
      X(IS) = TEMP
      TEMP = Y(L)
      Y(L) = Y(IS)
      Y(IS) = TEMP
      TEMP = NXY(L)
      NXY(L) = NXY(IS)
      NXY(IS) = TEMP
      TEMP = NUM(L)
      NUM(L) = NUM(IS)
      NUM(IS) = TEMP
C
C  COMPUTE TRAIL NUMBER
  212 ITRA(IS) = (NXY(IS)-NUM(IS)*1000)/100
      PRINT 834,RA,PMRA,DEC,PMDC,CATNO,MTYP,MPT,ITRA(IS),PL
      NCT=NCT+1
      PMRA = PMRA*1.E-02
      PMDC = PMDC*1.E-02
C
C  UPDATE TO YEAR OF OBSERVATION
      XL= COS(RA)*COS(DEC)
      YL= SIN(RA)*COS(DEC)
      Z = SIN(DEC)
      PMX = -PMRA*COS(DEC)*SIN(RA) - PMDC*COS(RA)*SIN(DEC)
      PMY =  PMRA*COS(DEC)*COS(RA) - PMDC*SIN(RA)*SIN(DEC)
      PMZ =  PMDC*COS(DEC)
      PMSQ = PMRA*PMRA*COS(DEC)*COS(DEC) + PMDC*PMDC
      PMXD = -XL* PMSQ
      PMYD = -YL* PMSQ
      PMZD = -Z * PMSG
      TSQ = .5*R(18)*R(18)
      Q(1) = XL+ PMX*R(18) + PMXD*TSQ
      Q(2) = YL+ PMY*R(18) + PMYD*TSQ
      Q(3) = Z + PMZ*R(18) + PMZD*TSQ
      T = R(18)/100.
      TSQ = T * T
      TCU = TSQ * T
      DELTA = 2304.948*T + .302*TSQ + .018*TCU
      ZETA = (DELTA + .791*TSQ) * RSEC
      DELTA = DELTA * RSEC
      THETA = (2004.2555*T - .426*TSG - .042*TCU) * RSEC
      DC(1,1) =  COS(DELTA)*COS(THETA)*COS(ZETA) - SIN(DELTA)*SIN(ZETA)
      DC(1,2) = -SIN(DELTA)*COS(THETA)*COS(ZETA) - COS(DELTA)*SIN(ZETA)
      DC(1.3) = -SIN(THETA)*COS(ZETA)
      DC(2,1) =  COS(DELTA)*COS(THETA)*SIN(ZETA) + SIN(DELTA)*COS(ZETA)
      DC(2,2) = -SIN(DELTA)*COS(THETA)*SIN(ZETA) + COS(DELTA)*COS(ZETA)
      DC(2,3) = -SIN(THETA)*SIN(ZETA)
      DC(3,1) =  COS(DELTA)*SIN(THETA)
      DC(3,2) = -SIN(DELTA)*SIN(THETA)
      DC(3,3) =  COS(THETA)
      DO 215 I=1,3
        QQ(I) = 0.
        DO 215 k=1,3
          QQ(I) = QQ(I) + DC(I,K)*Q(K)
  215 CONTINUE
      CALL ANGLE(QQ(2),QQ(1),RTASC)
      SQ = SQRT(QQ(1)*QQ(1) + QQ(2)*QQ(2))
      DECL = ATAN(QQ(3)/SQ)
C
C  MATCH WITH UPDATING PARAMETERS
      DO 222 I=1,JJ
      IF(ITRL(I).EQ.ITRA(IS)) GO TO 225
  222 CONTINUE
      PRINT 906,ITRA(IS)
      NCT=NCT+3
      IF(MTEST.EQ.0) GO TO 195
      GO TO 260
C
C  UPDATE TO TIME OF OBSERVATION
  225 ASTAR = RTASC + FFR(I) + TAU(I)*PMRA + GSMR(I)*TAN(DECL)*SIN(RTASC
     1  +GCPR(I)) + HSMR(I)*SIN(RTASC+HCPR(I))/COS(DECL)
      D = DECL + TAU(I)*PMDC + HSMR(I)*SIN(DECL)*COS(RTASC+HCPR(I)) +
     1  GSMR(I)*COS(RTASC+GCPR(I)) + SMIR(I)*COS(DECL)
      ALS(IS) = ASTAR
      DLS(IS) = D
C
C  ADJUST FOR DIURNAL ABERRATION
      IDA = 1
      SLAT = SIN(PHI)
      CLAT = COS(PHI)
  230 HPR = STR(I) - ASTAR
      SHPR = SIN(HPR)
      CHPR = COS(HPR)
      SNDL = SIN(D)
      CSDL = COS(D)
      IF(IDA.EQ.2) GO TO 235
      ASTAR = ASTAR + (.0213*RTSEC*CLAT*CHPR)/CSDL
      D = D + .32*RSEC*CLAT*SHPR*SNDL
      IDA = 2
      GO TO 230
C
C  ADJUST FOR REFRACTION
  235 CSZ = SNDL*SLAT + CSDL*CHPR*CLAT
      IF(ABS(CSZ).LT.1.) GO TO 238
      REF = 0.
      GO TO 240
  238 SNZ = SQRT(1.-CSZ*CSZ)
      TNZ = SNZ/CSZ
      SNQ = SHPR*CLAT/SNZ
      CSQ = (SLAT - SNDL*CSZ)/(CSDL*SNZ)
      RM = AP0*TNZ - AP1*TNZ**3 + AP2*TNZ**5
      PABAR = R(20)*(1.-.00264*COS(2.*PHI) - 2.*R(19)/PCON)
      REF = RM*PABAR*(1.+BETA*T0)/(P0*(1.+BETA*R(21)))
  240 DP = D + REF*CSQ
      SDP = SIN(DP)
      CDP = COS(DP)
      APSTR = ASTAR + REF*SNQ/CDP
C
C  COMPUTE STANDARD COORDINATES
      HPR = STR(I) - APSTR
      CHPR = COS(HPR)
      CSZ = SDP*SLAT + CDP*CHPR*CLAT
      PSI(IS) = (CLAT*SDP-SLAT*CDP*CHPR)/CSZ
      ADA(IS) = -CDP*SIN(HPR)/CSZ
      CATNA(IS) = CATNO
C
C  TEST FOR END OF DATA
      IS = IS+1
      IF(IS.GT.ISR) GO TO 250
      IF(MTEST.EQ.0) GO TO 195
      GO TO 260
  250 IF(MTEST.EQ.1) GO TO 260
      CALL DBUF(1)
      DECODE (80,17,BUFF(INBUF)) MTEST
      GO TO 250
C
C  OUTPUT STANDARD COORDINATES (OPTIONAL)
  260 IS = IS-1
      IPAGE = IPAGE+1
      PRINT 829,IPAGE
      PRINT 823
      PRINT 803,EV,PL
      IF(IS.EQ.ISR) GO TO 262
      IS1 = IS+1
      PRINT 907
      DO 261 I=IS1,ISR
      PRINT 908,NXY(I)
  261 CONTINUE
  262 PRINT 825
      PRINT 835
      DO 265 I=1,IS
      IF(IPUN(3).EQ.1) GO TO 263
      CALL OBUF(1)
      ENCODE (80,22,BUF(IBUF)) PSI(IS),ADA(IS),CATNA(IS),NXY(IS),PL
  263 PRINT 836,PSI(I),ADA(I),CATNA(I),NXY(I),PL
  265 CONTINUE
C
C  COMPUTE APPROXIMATE CAMERA ORIENTATION USING PLATE CONSTANT METHOD
      IPAGE=IPAGE+1
      PRINT 837,IPAGE
      PRINT 803,EV,PL
      PRINT 838,PHILH,PHILM,PHILS,XLAMLH,XLAMLM,XLAMLS
      NCT = 13
      N = IS
      IC = 0
      K1 = N
  273 DO 275 I=1,96
      F(I) = 0.
  275 CONTINUE
      IF(NCT.LE.37) GO TO 278
      IPAGE = IPAGE+1
      PRINT 837,IPAGE
      PRINT 823
      PRINT 803,EV,PL
      NCT = 7
  278 PRINT 825
      PRINT 839,N
      NCT = NCT+6
      M = 2*N
      J =-1
      DO 293 I=1,N
      J = J+2
      S(J,1) = PSI(I)
      S(J+1,2) = PSI(I)
      S(J,3) = ADA(I)
      S(J+1,4) = ADA(I)
      S(J,5) = 1.0
      S(J+1,6) □ 1.0
      S(J,7) = -PSI(I)* X(I)
      S(J+1,7) □ -PSI(I)* Y(I)
      S(J,8) = -ADA(I)* X(I)
      S(J+1,8) = -ADA(I)* Y(I)
      S(J,9) = X(I)
      S(J+1,9) □ Y(I)
      S(J,2)=S(J+1,1)=S(J+1,3)=S(J,4)=S(J+1,5)=S(J,6)=0.
  293 CONTINUE
      DO 295 I=1,8
      DO 295 J=1,9
      DO 295 L=1,M
      F(I,J)=F(I,J)+S(L,I)*S(L,J)
  295 CONTINUE
      CALL ERWIN (8,1,F.IS,8,12)
      IF(IS.GT.0) GO TO 300
      PRINT 909
      GO TO 100
  300 A1 = F(1,9)
      A2 = F(2,9)
      B1 = F(3,9)
      B2 = F(4,9)
      C1 = F(5,9)
      C2 □ F(6,9)
      A0 = F(7,9)
      B0 = F(8.9)
      AP = A1*B2-A2*B1
      A0P =(A2*B0 -A0*B2)/AP
      A2P=A0*C2-A2
      B0P =(A0*B1 - A1*B0)/AP
      B2P=A0*C1-A1
      ALPHA =ATAN (A0)
      OMEGA =ATAN (B0/ (SQRT (A0**2 +1.0)))
      CALL ANGLE (A2P,B2P,RKAPPA)
      XY = A0**2 + B0**2 + 1.0
      XP =(A0*A1 + B0*B1 + C1)/XY
      YP =(A0*A2 + B0*B2 + C2)/XY
      C=((AP**2*(1.0+A0P*XP+B0P*YP)**3)/(1.0+A0P*C1+B0P*C2))**.25
      GALPHA = ALPHA * RDTOGR
      GOMEGA = OMEGA * RDTOGR
      GKAPPA □ RKAPPA * RDTOGR
      PRINT 840,GALPHA,GOMEGA,GKAPPA,XP,YP,C
      NCT=NCT+6
C
C  COMPUTE STANDARD COORDINATES FROM CAMERA ORIENTATION,
C   AND FIND RESIDUALS
      SINALP = SIN (ALPHA)
      COSALP = COS (ALPHA)
      SINOME = SIN (OMEGA)
      COSOME = COS (OMEGA)
      SINKAP = SIN (RKAPPA)
      COSKAP = COS (RKAPPA)
      A1 = -COSALP*COSKAP + SINALP*SINOME*SINKAP
      B1 = -COSOME*SINKAP
      C1 = SINALP*COSKAP + COSALP*SINOME*SINKAP
      A2 = -COSALP*SINKAP - SINALP*SINOME*COSKAP
      B2 =  COSOME*COSKAP
      C2 =  SINALP*SINKAP - COSALP*SINOME*COSKAP
      D = SINALP*COSOME
      E = SINOME
      FF= COSALP*COSOME
      PRINT 8411
      DO 305 I=1,N
      DE =  (X(I)-XP)*C1 + (Y(I)-YP)*C2 + C*FF
      TI = ((X(I)-XP)*A1 + (Y(I)-YP)*A2 + C*D)/DE
      EI = ((X(I)-XP)*B1 + (Y(I)-YP)*B2 + C*E)/DE
      VS(I,1) = PSI(I) - TI
      VS(I,2) = ADA(I) - EI
      PRINT 841,TI,EI,VS(I,1),VS(I,2),NXY(I)
  305 CONTINUE
      NCT = NCT+N+7
C
C  TEST RESIDUALS AGAINST TOLERANCE
      VMAX = ABS(VS(1,1))
      JK = 1
      ICT = 0
      DO 306 I=1,N
      ISIG = 0
      DO 306 J=1,2
      IF(ABS(VS(I,J)).LE.CRI) GO TO 304
      IF(ISIG.EQ.0) ICT = ICT+1
      ISIG = 1
  304 IF(ABS(VS(I,J)).LE.VMAX) GO TO 306
      VMAX = ABS(VS(I,J))
      JK = I
  306 CONTINUE
      IF(ICT.EQ.0) GO TO 310
      IF(N.EQ.4) GO TO 400
      IF(ICT.GT.1) GO TO 309
C
C   IF RESIDUAL EXCEEDS TOLERANCE, REMOVE STAR AND RECOMPUTE
      PRINT 842,NXY(JK)
      NCT = NCT+4
      N = N-1
      K1 = N
      IF(JK.EQ.N+1) GO TO 273
      DO 308 I=JK,N
      PSI(I) = PSI(I+1)
      ADA(I) = ADA(I+1)
      CATNA(I) = CATNA(I+1)
      X(I) = X(I+1)
      Y(I) □ Y(I+1)
      NXY(I) = NXY(I+1)
      NUM(I) = NUM(I+1)
      ALS(I) = ALS(I+1)
      DLS(I) = DLS(I+1)
      ITRA(I) = ITRA(I+1)
  308 CONTINUE
      GO TO 273
C
C  IF MANY RESIDUALS EXCEED TOLERANCE, REMOVE EACH STAR IN TURN
C   AND RECOMPUTE
  309 IC = IC+1
      JK = K1-IC+1
      IF(JK.EQ.0) GO TO 400
      PRINT 842,NXY(JK)
      NCT = NCT+4
      N = K1-1
      IF(JK.EQ.K1) GO TO 273
      TEMP □ PSI(JK)
      PSI(JK) □ PSI(K1)
      PSI(K1) = TEMP
      TEMP = ADA(JK)
      ADA(JK) □ ADA(K1)
      ADA(K1) = TEMP
      TEMP □ CATNA(JK)
      CATNA(JK) = CATNA(K1)
      CATNA(K1) □ TEMP
      TEMP = X(JK)
      X(JK) = X(K1)
      X(K1) = TEMP
      TEMP = Y(JK)
      Y(JK) = Y(K1)
      Y(K1) = TEMP
      TEMP □ NXY(JK)
      NXY(JK) □ NXY(K1)
      NXY(K1) = TEMP
      TEMP □ NUM(JK)
      NUM(JK) = NUM(K1)
      NUM(K1) □ TEMP
      TEMP □ ALS(JK)
      ALS(JK) = ALS(K1)
      ALS(K1) = TEMP
      TEMP □ DLS(JK)
      DLS(JK) = DLS(K1)
      DLS(K1) = TEMP
      TEMP = ITRA(JK)
      ITRA(JK) = ITRA(K1)
      ITRA(K1) = TEMP
      GO TO 273
C
C  OUTPUT PRELIMINARY CAMERA ORIENTATION, LOCAL
  310 CALL OBUF(1)
      ENCODE (80,50,BUF(IBUF)) IDUM2
      CALL OBUF(1)
      ENCODE (80,23,BUF(IBUF)) GALPHA,GOMEGA,GKAPPA,PL
      CALL OBUF(1)
      ENCODE (80,23,BUF(IBUF)) XP,YP,C,PL
      CALL OBUF(1)
      ENCODE (80,50,BUF(IBUF)) IDUM2
C
C  ROTATE CAMERA ORIENTATION TO REFERENCE STATION
      PHIL = PHILH*RDG + PHILM*RMIN + PHILS*RSEC
      PHIR = PHIRH*RDG + PHIRM*RMIN + PHIRS*RSEC
      XLAML = XLAMLH*RDG + XLAMLM*RMIN + XLAMLS*RSEC
      XLAMR = XLAMRH*RDG + XLAMRM*RMIN + XLAMRS*RSEC
      SINPHL = SIN (PHIL)
      COSPHL = COS (PHIL)
      SINPHR = SIN (PHIR)
      COSPHR = COS (PHIR)
      SINLAL = SIN (XLAML)
      COSLAL = COS (XLAML)
      SINLAR = SIN (XLAMR)
      COSLAR = COS (XLAMR)
      A1P = -COSLAL*SINPHL*A1 + SINLAL*B1 + COSLAL*COSPHL*C1
      B1P = -SINLAL*SINPHL*A1 - COSLAL*B1 + SINLAL*COSPHL*C1
      A2P = -COSLAL*SINPHL*A2 + SINLAL*B2 + COSLAL*COSPHL*C2
      B2P = -SINLAL*SINPHL*A2 - COSLAL*B2 + SINLAL*COSPHL*C2
      C2P = COSPHL*A2 + SINPHL*C2
      DP = -COSLAL*SINPHL*D + SINLAL*E + COSLAL*COSPHL*FF
      EP = -SINLAL*SINPHL*D - COSLAL*E + SINLAL*COSPHL*FF
      FP = COSPHL*D + SINPHL*FF
      B1PP = SINLAR*A1P - COSLAR*B1P
      A2PP = -SINPHR*COSLAR*A2P - SINPHR*SINLAR*B2P + COSPHR*C2P
      B2PP =  SINLAR*A2P - COSLAR*B2P
      C2PP =  COSPHR*COSLAR*A2P + COSPHR*SINLAR*B2P + SINPHR*C2P
      DPP = -SINPHR*COSLAR*DP - SINPHR*SINLAR*EP + COSPHR*FP
      EPP = SINLAR*DP - COSLAR*EP
      FPP = COSPHR*COSLAR*DP + COSPHR*SINLAR*EP + SINPHR*FP
      IF(EPP.EQ.0.) GO TO 312
      COSOMR = SQRT(1.-EPP*EPP)
      OMEGAR = ATAN(EPP/COSOMR)*RDTOGR
      SINALR = DPP/COSOMR
      COSALR = FPP/COSOMR
      CALL ANGLE(SINALR,COSALR,ALPHAR)
      ALPHAR = ALPHAR*RDTOGR
      SINKAR = -B1PP/COSOMR
      COSKAR =  B2PP/COSOMR
      CALL ANGLE (SINKAR,COSKAR,RKAPPR)
      RKAPPR = RKAPPR*RDTOGR
      GO TO 315
  312 RKAPPR = 0.
      SIGN = 1.
      IF(EPP.LT.0.) SIGN = -1.
      OMEGAR = 100.*SIGN
      SINALR = -A2PP/SIGN
      COSALR = -C2PP/SIGN
      CALL ANGLE (SINALR,COSALR,ALPHAR)
      ALPHAR = ALPHAR*RDTOGR
C
C  PRINT ROTATED PRELIMINARY CAMERA ORIENTATION
  315 IF(NCT.LE.56) GO TO 318
      IPAGE = IPAGE+1
      PRINT 837,IPAGE
      PRINT 823
      PRINT 803,EV,PL
      NCT = 7
  318 PRINT 843
      PRINT 840,ALPHAR,OMEGAR,RKAPPR,XP,YP,C
C
C  OUTPUT SINGLE CAMERA HEADERS (REGULAR, OR PRELIMINARY OPTIONAL)
      IPAGE = IPAGE+1
      PRINT 844,IPAGE
      CALL OBUF(1)
      ENCODE (80,50,BUF(IBUF)) IDUM3
      IF(IPRE.EQ.0) GO TO 320
      CALL OBUF(1)
      ENCODE (80,36,BUF(IBUF)) BLANK
      CALL OBUF(1)
      ENCODE (80,37,BUF(IBUF)) N
      PRINT 853
      GO TO 325
  320 CALL OBUF(1)
      ENCODE (80,24,BUF(IBUF)) BLANK
      CALL OBUF(1)
      ENCODE (80,25,BUF(IBUF)) IDUMO
      PRINT 845
  325 CALL OBUF(1)
      ENCODE (80,26,BUF(IBUF)) TEM,PRES
      CALL OBUF(1)
      ENCODE (80,27,BUF(IBUF)) EV,DATE
      CALL OBUF(1)
      ENCODE (80,28,BUF(IBUF)) PL,CAM,UNIT
      CALL OBUF(1)
      ENCODE (80,29,BUF(IBUF)) COMM,OPER
      PRINT 846,TEM,PRES,EV,DATE,PL,CAM,UNIT,COMM,OPER
      CALL OBUF(1)
      ENCODE (80,30,BUF(IBUF)) ISTA1,(AN(I),I=1,3),ISIGS,PHILH,PHILM,
     1  PHILS,XLAMLH,XLAMLM,XLAMLS,EL1
      PRINT 847,ISTA1,(AN(I),I=1,3),ISIGS,PHILH,PHILM,PHILS,XLAMLH,
     1  XLAMLM,XLAMLS,EL1
      IF(IPRE.EQ.1) GO TO 328
      CALL OBUF(1)
      ENCODE (80,30,BUF(IBUF)) ISTA2,(AN(I),I=4,6),ISIGR,PHIRH,PHIRM,
     1  PHIRS,XLAMRH,XLAMRM,XLAMRS,EL2
      PRINT 847,ISTA2,(AN(I),I=4,6),ISIGR,PHIRH,PHIRM,PHIRS,XLAMRH,
     1  XLAMRM,XLAMRS,EL2
      CALL OBUF(1)
      ENCODE (80,31,BUF(IBUF)) XPC,YPC
      PRINT 848,XPC,YPC
  328 CALL OBUF(1)
      ENCODE (80,32,BUF(IBUF)) GALPHA,GOMEGA,GKAPPA
      CALL OBUF(1)
      ENCODE (80,32,BUF(IBUF)) ZERO,XP,YP
      CALL OBUF(1)
      ENCODE (80,32,BUF(IBUF)) C,C,ZERO
      PRINT 849,GALPHA,GOMEGA,GKAPPA,ZERO,XP,YP,C,C,ZERO
      CALL OBUF(1)
      ENCODE (80,32,BUF(IBUF)) ZERO,ZERO,ZERO
      CALL OBUF(1)
      ENCODE (80,32,BUF(IBUF)) ZERO,ZERO,ZERO
      CALL OBUF(1)
      ENCODE (80,32,BUF(IBUF)) ZERO
      PRINT 849,ZERO,ZERO,ZERO,ZERO,ZERO,ZERO,ZERO
      IF(IPRE.EQ.0) GO TO 330
      CALL OBUF(1)
      ENCODE (80,38,BUF(IBUF)) IDUM0
      CALL OBUF(1)
      ENCODE (80,39,BUF(IBUF)) IDUM1
      PRINT 854
      GO TO 332
  330 CALL OBUF(1)
      ENCODE (80,33,BUF(IBUF)) BLANK
      PRINT 851
  332 CALL OBUF(1)
      ENCODE (80,34,BUF(IBUF)) BLANK
      CALL OBUF(1)
      ENCODE (80,35,BUF(IBUF)) BLANK
      PRINT 852
      IF(IPRE.EQ.0) GO TO 335
      CALL OBUF(1)
      ENCODE (80,40,BUF(IBUF)) (NUM(I),ITRA(I),I=1,N)
      PRINT 855,(NUM(I),ITRA(I),I=1,N)
      DO 333 I=1,N
      CALL OBUF(1)
      ENCODE (80,41,BUF(IBUF)) ALS(I),DLS(I),NUM(I),ITRA(I),CATNA(I)
      PRINT 856,ALS(I),DLS(I),NUM(I),ITRA(I),CATNA(I)
      IT = ITRA(I)
      CALL OBUF(1)
      ENCODE (80,42,BUF(IBUF)) X(I),Y(I),NXY(I),NUM(I),ITRA(I),STL(IT)
      PRINT 857,X(I),Y(I),NXY(I),NUM(I),ITRA(I),STL(IT)
  333 CONTINUE
  335 PRINT 858
      CALL OBUF(1)
      ENCODE (80,50,BUF(IBUF)) IDUM3
C
C  OUTPUT PLATE DATA REDUCTION HEADERS
      CALL OBUF(1)
      ENCODE (80,50,BUF(IBUF)) IDUM5
      CALL OBUF(1)
      ENCODE (80,43,BUF(IBUF)) EV,PL
      PRINT 859,EV,PL
      CALL OBUF(1)
      ENCODE (80,44,BUF(IBUF)) ICOM,(RL(I),I=1,4)
      DO 340 K=5,30,5
      MQ = K+4
      CALL OBUF(1)
      ENCODE (80,32,BUF(IBUF)) (RL(I),I=K,MQ)
  340 CONTINUE
      CALL OBUF(1)
      ENCODE (80,32,BUF(IBUF)) GALPHA,GOMEGA,GKAPPA
      CALL OBUF(1)
      ENCODE (80,32,BUF(IBUF)) XP,YP,C
      CALL OBUF(1)
      ENCODE (80,45,BUF(IBUF)) ISTA1,(AN(I),I=1,3),ISIGS,PHILH,PHILM,
     1  PHILS,XLAMLH,XLAMLM,XLAMLS,EL1
      PRINT 860,ICOM,(RL(I),I=1,34)
```

```
    PRINT 849,GALPHA,GOMEGA,GKAPPA,XP,YP,C
    PRINT 861,ISTA1,(AN(I),I=1,3),ISIGS,PHILH,PHILM,PHILS,XLAMLH,
   1  XLAMLM,XLAMLS,EL1
    CALL OBUF(1)
    ENCODE (80,50,BUF(IBUF)) IDUM5
    CALL OBUF(0)
    GO TO 100
400 PRINT 911
    GO TO 100
  1 FORMAT(18X,A3,A9,5A10)
  2 FORMAT(2X,A5,5X,A4,6X,A8,1X,I1,4X,3(1X,I1),A4,5X,I1)
  3 FORMAT(3X,A4,3A8,I1,1X,2F4.0,F8.4,F5.0,F4.0,F8.4,E14.8)
  4 FORMAT(4X,A8,6X,A3,6X,A10,A7,6X,3A8)
  5 FORMAT(5X,A10,6X,A10,2(5X,F7.4))
  6 FORMAT(4X,A8,A6,9X,2A10,8X,A4,9X,A5)
  7 FORMAT(5X,I3,2(1X,I2),6X,I3,2(1X,I2),7XA10,9X,2A10)
  8 FORMAT(8X,3(F2.0,1X),6X,F4.0,6X,2(F2.0,1X),F4.1,10X,F4.0,6X,
   1  2(F2.0,1X),F4.1)
  9 FORMAT(9X,F2.0,9X,F5.1,11X,2(F2.0,3X),5X,F2.0,3X,F2.0)
 10 FORMAT(3X,F3.0,2(1X,F2.0),6X,F3.0,2(1X,F2.0),12X,F3.0,1X,F2.0,
   1  1X,F8.5)
 11 FORMAT(1X,78H
   1                           ,I1)
 12 FORMAT(5(1X,E14.8))
 13 FORMAT(11X,I1,3X,4(1X,E14.8))
 14 FORMAT(I4,I1,I6,2F6.6,56X,I1)
 15 FORMAT(63X,I1,3X,A4,3I2)
 16 FORMAT(4(F11.8,1X),2X,A6,2X,I2,I3,I1,3X,A4,8X,I1)
 17 FORMAT(79X,I1)
 20 FORMAT(*B        JOB STEP *,A3,A9,5A10)
 21 FORMAT(2E14.7,30X,I2,I6,1X,A4,8X,I1)
 22 FORMAT(2E14.7,2X,A6,14X,I6,1X,A4)
 23 FORMAT(3(1X,E14.7),2X,A4,20X,I2)
 24 FORMAT(A4,36HSINGLE CAMERA ORIENTATION   PROG 377)
 25 FORMAT( I1,4X,1H0,4X,1H1,4X,1H1,4X,1H0,5X,1H0,3X,1H0,4X,1H2,4X,1H1
   1,5X,1H1,3X,1H0,4X,1H0,3X,2H10)
 26 FORMAT(20H+100000-06+100000-01,2A10,40H+650000-01+300000-02+126000
   1+00+100000-06)
 27 FORMAT(A5,1X,A10,A7)
 28 FORMAT(A4,2X,A8,6X,A3)
 29 FORMAT(A8,2X,3A8,17X,5H3 6 9)
 30 FORMAT(A4,2X,3A8,2X,I1,2X,2F3.0,F8.4,3X,2F3.0,F8.4,3X,F11.4)
 31 FORMAT(7X,F7.4,3X,F7.4)
 32 FORMAT(5(1X,E14.7))
 33 FORMAT(A1,79H101          +001515+1       +211616+1       +210101
   1       +000101          +00)
 34 FORMAT(36H 26500000+01 26500000+01 00000000+00,A1)
 35 FORMAT(36H 10000000+01 10000000+01 00000000+00,A1)
 36 FORMAT(A4,33HSINGLE CAMERA ORIENTATION PRE RUN)
 37 FORMAT(1H1,4X,1H0,2(4X,1H1),4X,1H0,4X,I2, 3X,1H1,2(4X,1H0),5X,1H2,.
   1  3X,1H0,4X,1H5,4X,1H0)
 38 FORMAT(I1,*404 10000000+210909 10000000+211010 10000000+211111 100
   100000+211212 10000000+21*)
 39 FORMAT(I2, *13 10000000+211414 10000000+211515 10000000+211616 100
   100000+210101 00000000+00*)
 40 FORMAT(10(1X,I2,I1))
 41 FORMAT(    2(E14.7),30H .1000E+01 .1000E+01 .0000E+00,4X,I3,I1,4X,
     11H1,1X,A6,1X,1H0)
  42 FORMAT( 2F12.10 ,30H .1000E+01 .1000E+01 .0000E+00,I6,2X,I3,I1,1X,
     1F10.8,2X,1H0)
  43 FORMAT(A5,1X,A4)
  44 FORMAT(11X,I1,3X,4(1X,E14.7))
  45 FORMAT(3X,A4,3A8,I1,1X,2(F3.0,1H.),F8.4,F4.0,1H.,F3.0,1H.,F8.4,
     1  E14.7)
  50 FORMAT(78X,I2)
 801 FORMAT(1H1,30(/),60X,*   JOB STEP *,A3///50X,A9,5A10)
 802 FORMAT(1H1,72X,5HPAGE ,I2/24X,*PRELIMINARY DATA REDUCTION *)
 803 FORMAT(//23X,6HEVENT ,A5,5X,6HPLATE ,A4 /)
 804 FORMAT(31X,10HCOMP. NO. ,A8)
 805 FORMAT(//19X 5HPRINT,16X,5HPUNCH,9X,5HPUNCH/19X,4HDATA,18X,
     1  18HD.H.  PUNCH  STAND/18X,6HRECORD,17X,18HSAT.  STARS  COOR./
     2  21X,I1,14X,3(6X,I1))
 806 FORMAT(//5X,6HCAMERA,6X,4HUNIT,13X,4HDATE,19X,8HOPERATOR/4X,A8,
     1  6X,A3,6X,A10,A7,6X,3A8)
 807 FORMAT(//4X,12H TEMPERATURE,6X,9H PRESSURE,7X,3H XP,9X,3H YP/5X,
     1  A10,6X,A10,2(5X,F7.4))
 808 FORMAT(//3X,5HLOCAL,8X,5HLOCAL)
 809 FORMAT(//4X,4HREF.,9X,4HREF.)
 810 FORMAT( 2X,7HSTA.NO.,5X,9HSTA. NAME,14X,8HLATITUDE,8X,9HLONGITUDE,
     17X,9HELEVATION/3X,A4,3A8,I1,1X,2F4.0,F8.4,F5.0,F4.0,F8.4,E14.7)
 811 FORMAT(1H1,72X,5HPAGE ,I2 / 24X,32HSATELLITE  TRIANGULATION  RECOR
     1D  //)
 812 FORMAT(25X,7HEVENT  ,A5,8X,7HPLATE  ,A4//32X,10HSATELLITE ,A8,A6//
     1  31X,5HDATE ,A10,A7///)
 813 FORMAT(33X,12HSTATION DATA//10X,6HNUMBER,22X,4HNAME,20X,8HLOCATION
     1  /10X,A4,14X,3A8,5X,2A10//9X,8HLATITUDE,18X,9HLONGITUDE,17X,
     2  9HELEVATION/6X,2F3.0,F8.4,12X,2F3.0,F8.4,14X,F11.4///)
 814 FORMAT(    34X,11HCAMERA DATA //  7X,11HUNIT NUMBER,16X,11HLENS NU
     1MBER,14X,13HLENS APERTURE / 11X,A3,22X,A8,19X,A4 // 5X,15HELEVATIO
     2N ANGLE,13X,14HAZIMUTH(NORTH),13X,11HSLANT RANGE / 8X,3I3,18X,3I3,
     318X,A5,1X,2HKM // 7X,11HTEMPERATURE,18X,8HPRESSURE,19X,7HWEATHER /
     4  6X,A10,1X,1HC,16X,A10,1X,2HMM,10X,2A10///)
 815 FORMAT(35X,9HTIME DATA//6X,5HSTART,21X,15HSATELLITE TRAIL,21X,
     1  4HTIME/1X,15HPRE-CALIBRATION,2X,19HFIRST SATELLITE NO.,F4.0,
     2  6H TIME ,F2.0,2HH ,F2.0,2HM ,F4.1,1HS,5X,10HCORRECTION/3X,F2.0,
     3  2HH ,F2.0,2HM ,F2.0,1HS,5X,18HLAST SATELLITE NO.,F4.0,6H TIME ,
     4  F2.0,2HH ,F2.0,2HM ,F4.1,1HS,2X,F2.0,2HH ,F2.0,2HM ,F8.5,1HS/)
 816 FORMAT(34X,11HACS MONITOR/9X,11HSTAR FORMAT,14X,3HPRE,4X,4HPOST,
     1  13X,10HDAY NUMBER/13X,F2.0,14X,5HOPEN ,F2.0,3H MS,2X,F2.0,3H MS,
     2  15X,F5.1/28X,6HCLOSE ,F2.0,3H MS,2X,F2.0,3H MS///)
 817 FORMAT(22X,15HRIGHT ASCENSION,5X,11HDECLINATION/16X,2(9X,F3.0)///
     1  30X,17HPOLAR COORDINATES/34X,1HX,9X,1HY/27X,2(3X,F7.4))
 818 FORMAT(/// 36X,7HREMARKS /)
 819 FORMAT(1H1,72X,5HPAGE ,I2/24X,*PRELIMINARY COMPARATOR REDUCTION*)
 820 FORMAT(34X,9HCOMP. NO.,I3)
 821 FORMAT(/37X,5HINPUT/)
 822 FORMAT(24X,5HPOINT,4X,5HPOINT/24X,5HCLASS,3X,6HNUMBER,6X,1HX,8X,
     1  1HY)
 823 FORMAT(36X,7H(CONT.))
 824 FORMAT(25X,I2,2X,3(3X,I6))
 825 FORMAT(//37X,6HOUTPUT /)
 826 FORMAT(//26X,10HDRILL HOLE,7X,10HDRILL HOLE/27X,9HDIST. 1-3,8X,
     1  9HDIST. 2-4/24X,E14.7,3X,E14.7//28X,7HTRANS X,10X,7HTRANS Y/
 827 FORMAT(55X,5HPOINT,3X,5HPOINT/9X,2HCX,16X,2HCY,26X,20HCLASS  NUMBE
    1R  PLATE)
 828 FORMAT(    4X,E14.7,4X,E14.7,20X,I2,4X,I6,3X,A4)
 829 FORMAT(1H1,72X,5HPAGE ,I2 / 26X,26HPRELIMINARY STAR REDUCTION )
 830 FORMAT(     60X,18HTRAIL  PLATE  DATE / 6X,7HLAT-DEG,8X,7HLAT-MIN,8
    1X,7HLAT-SEC / 6X,7HLST-HR ,8X,7HLST-MIN,8X,7HLST-SEC / 8X,5HG-HR ,
    28X,7H  G-MIN,8X,7H  G-SEC/8X,4HH-HR,11X,5HH-MIN,10X,5HH-SEC/9X,
    3  1HF,14X,1HG,14X,1HH/9X,1HI,13X,3HTAN,10X,7HDELTA T/5X,9HELEVATIO
    4N,6X,8HPRESSURE,6X,11HTEMPERATURE)
 831 FORMAT(/62X,I1,5X,A4,1X,3I2)
 832 FORMAT(1X,3(1X,E14.7))
 833 FORMAT(//5X,5HRIGHT,43X,4HBOSS,4X,5HCLASS,3X,5HPOINT/3X,9HASCENSIO
    1N,5X,5HPM RA,4X,11HDECLINATION,4X,6HPM DEC,5X,6HNUMBER,2X,6HNUMBER
    2 ,2X,6HNUMBER,1X,5HPLATE/)
 834 FORMAT(1X,4(1X,F11.8),3X,A6,4X,I2,4X,I3,I1,4X,A4)
 835 FORMAT(9X,21HSTANDARD  COORDINATES,23X,4HBOSS,8X,5HPOINT/9X,3HPSI,
    1  15X,3HETA,22X,6HNUMBER,6X,6HNUMBER,5X,5HPLATE/)
 836 FORMAT(4X,E14.7,4X,E14.7,16X,A6,6X,I6,6X,A4)
 837 FORMAT(1H1,72X,5HPAGE ,I2/24X,30HPRELIMINARY CAMERA ORIENTATION )
 838 FORMAT(/20X,8HLATITUDE,23X,9HLONGITUDE/17X,2F3.0,F8.4,18X,2F3.0,
    1  F8.4///)
 839 FORMAT(  / 13X,45HPRELIMINARY CAMERA ORIENTATION COMPUTED WITH ,I2
    1,6H STARS)
 840 FORMAT(     / 21X,5HALPHA,10X,5HOMEGA,10X,5HKAPPA / 15X,3(1X,E14.7)
    1 // 22X,2HXP,13X,2HYP,14X,1HC/ 15X,3(1X,E14.7))
8411 FORMAT(///9X,*STANDARD COORDINATES AND RESIDUALS COMPUTED FROM ORI
    1ENTATION *//65X,6H POINT/10X,4H PSI,11X,4H ETA,8X,10H DELTA PSI,
    2  5X,10H DELTA ETA,2X,7H NUMBER)
 841 FORMAT(4X,4(1X,E14.7),1X,I6)
 842 FORMAT(///2X,60H* PRELIMINARY CAMERA ORIENTATION TO BE RECOMPUTED
    1WITH STAR ,I6,10H REMOVED *)
 843 FORMAT(  /// 20X,38HROTATED PRELIMINARY CAMERA ORIENTATION)
 844 FORMAT(1H1,72X,5HPAGE ,I2 /32X,14HS.C.O. HEADERS /)
 845 FORMAT(5X,36HSINGLE CAMERA ORIENTATION   PROG 377/1X,1H0,4X,1H0,
    1  2(4X,1H1),4X,1H0,5X,1H0,3X,1H0,4X,1H2,4X,1H1,5X,1H1,3X,1H0,4X,
    2  1H0,3X,2H10)
 846 FORMAT(1X,20H+100000-06+100000-01,2A10,40H+650000-01+300000-02+126
    1000+00+100000-06/1X,A5,1X,A10,A7/1X,A4,2X,A8,6X,A3/1X,A8,2X,
    2  3A8,17X,5H3 6 9)
 847 FORMAT(1X,A4,2X,3A8,2X,I1,2X,2F3.0,F8.4,3X,2F3.0,F8.4,3X,F11.4)
 848 FORMAT(8X,F7.4,3X,F7.4)
 849 FORMAT(2X,E14.7,1X,E14.7,1X,E14.7)
 851 FORMAT(1X,80H 101           +001515+1        +211616+1        +210101
    1        +000101         +00)
 852 FORMAT(1X,36H 26500000+01 26500000+01 00000000+00/1X,36H 10000000+
    101 10000000+01 00000000+00)
 853 FORMAT(5X,33HSINGLE CAMERA ORIENTATION PRE RUN/1X,1H1,4X,1H0,2(4X,
    1  1H1),4X,1H0,4X,I2,3X,1H1,2(4X,1H0),5X,1H2,3X,1H0,4X,1H5,4X,1H0)
 854 FORMAT(1X,80H0404 10000000+210909 10000000+211010 10000000+211111
    110000000+211212 10000000+21/1X,80H1313 10000000+211414 10000000+21
    21515 10000000+211616 10000000+210101 00000000+00)
 855 FORMAT(1X,10(1X,I2,I1))
 856 FORMAT(1X,2E14.7      ,30H .1000E+01 .1000E+01 .0000E+00,4X,I3,I1,4
    1X,1H1,1X,A6,1X,1H0)
 857 FORMAT(1X,2F12.10 ,30H .1000E+01 .1000E+01 .0000E+00,I6,2X,I3,I1,1
    1X,F10.8,2X,1H0)
 858 FORMAT(//25X,* PLATE DATA REDUCTION HEADERS */)
```

```
859 FORMAT(1X,A5,1X,A4)
860 FORMAT(12X,I1,3X,4(1X,E14.7)/(2X,E14.7,1X,E14.7,1X,E14.7,1X,E14.7,
   1  1X,E14.7))
861 FORMAT(4X,A4,3A8,I1,1X,2(F3.0,1H.),F8.4,F4.0,1H.,F3.0,1H.,F8.4,
   1  E14.7)
899 FORMAT(*1  JOB STEP ABORTED - INPUT DATA */(1X,8A10))
901 FORMAT(// 10X,50H*    *** WRONG REF. STATION USED - USE STATION NO.
   1 ,A4,8H ***    *)
902 FORMAT(/ 13X,31HFIRST 25 POINTS USED FOR POINT I6 /)
903 FORMAT(/25H DRILL HOLES OUT OF ORDER/)
904 FORMAT(/25X,* ONLY FIRST TEN STARS USED*/)
905 FORMAT(/25X,* ONLY FIRST TWELVE SATELLITES USED */)
906 FOPMAT(/25X,26NO HEADER CARDS FOR TRAIL I1/)
907 FORMAT(/12X,58H* NO RIGHT ASCENSION-DECLINATION FOR THE FOLLOWING
   1STARS * /)
908 FORMAT(37X,I6)
909 FORMAT(25X,* NO INVERSE FOR NORMAL MATRIX */)
910 FORMAT(//13X,* PLATE NUMBERS DO NOT AGREE*/* PRE SET    *,A4,10X,
   1   * UPDATING PARAMETERS    *,A4)
911 FORMAT(/* SOLUTION NOT REACHED FOR THIS SET OF PRE STARS *)
    END
```

```
      SUBROUTINE PLTCEN (XF,YF,XC,YC)
      DIMENSION XF(8),YF(8),QQ(4,3),RN(2,3)
      K = 0
      DO 165 J=1,5,4
        K = K+1
        QQ(K,1) = YF(J+2) - YF(J)
        QQ(K,2) = XF(J) - XF(J+2)
        QQ(K,3) = XF(J)*YF(J+2) - XF(J+2)*YF(J)
        K = K+1
        QQ(K,1) = YF(J+3) - YF(J+1)
        QQ(K,2) = XF(J+1) - XF(J+3)
        QQ(K,3) = XF(J+1)*YF(J+3) - XF(J+3)*YF(J+1)
  165   CONTINUE
      DO 166 J=1,2
        DO 166 K=1,3
          RN(J,K) = 0.
          DO 166 L=1,4
            RN(J,K) = RN(J,K) + QQ(L,J)*QQ(L,K)
  166       CONTINUE
      DET = RN(1,1)*RN(2,2) - RN(1,2)*RN(2,1)
      XC = (RN(1,3)*RN(2,2) - RN(1,2)*RN(2,3))/DET
      YC = (RN(1,1)*RN(2,3) - RN(1,3)*RN(2,1))/DET
      RETURN
      END
      SUBROUTINE ERWIN (N,NC,A,IS,NJ,NK)
      DIMENSION A(NJ,NK)
C     MODIFIED CHOLESKY
      IS = 1
        I = 1
      DO 10 J=1,N
        GO TO (5,3,1),I
    1   K = J-2
      DO 2 L=1,K
      DO 2 M=J,N
    2   A(J-1,M) □ A(J-1,M) - A(L,J-1)*A(L,M)
    3   K = J-1
      DO 4 L=1,K
    4   A(J,J) = A(J,J) - A(L,J)*A(L,J)
        I = 2
    5 IF(A(J,J)) 6, 6, 7
    6 IS = -1
      GO TO 22
    7    A(J,N+3) □ SQRT (A(J,J))
      DO 8 L=J,N
    8   A(J,L) = A(J,L)/A(J,N+3)
      DO 9 L=1,J
    9   A(L,J) = A(L,J)/A(J,N+3)
   10   I = I+1
C     INVERSION OF U
      DO 11 I=2,N
        J = I-1
      DO 11 K=1,J
   11   A(I,K) □ -A(K,I)
      DO 12 I=3,N
        J = I-2
      DO 12 K=1,J
        L = I-K-1
        M = I
      DO 12 IJ=1,K
        M = M-1
   12   A(I,L) = A(I,L) - A(I,M)*A(L,M)
C     COMPUTATION OF U INVERSE * U INVERSE TRANSPOSE
      DO 14 I=1,N
      DO 14 J=I,N
        A(I,N+4) = 0.
      DO 13 K=J,N
   13   A(I,N+4) = A(I,N+4) + A(K,I)*A(K,J)
   14   A(I,J) = A(I,N+4)
      DO 15 I=1,N
      DO 15 J=I,N
   15   A(I,J) □ A(I,J)/A(I,N+3)
      DO 16 I=1,N
      DO 16 J=1,I
   16   A(J,I) □ A(J,I)/A(I,N+3)
      DO 17 K=1,N
      DO 17 L=K,N
   17   A(L,K) = A(K,L)
C     COMPUTATION OF SOLUTIONS
      IF(NC) 18,22,18
   18 M  = N+NC
      I  = N+1
      DO 21 J=I,M
      DO 19 K=1,N
19      A(K,N+4) = 0.
      DO 20 K=1,N
      DO 20 L=1,N
20      A(K,N+4) = A(K,N+4) + A(K,L)*A(L,J)
      DO 21 K=1,N
21      A(K,J) = A(K,N+4)
22    RETURN
      END
      SUBROUTINE ANGLE (Y,X,A)
      PI □ 3.1415926535898
      IF(X) 30,20,10
   10 IF(Y.LT.0.) GO TO 13
      A □ ATAN(Y/X)
      GO TO 40
   13 A = 2.*PI + ATAN(Y/X)
      GO TO 40
   20 IF (Y) 23,22,21
   21 A = PI/2.
      GO TO 40
   22 A = 0.
      GO TO 40
   23 A = 3.*PI/2.
      GO TO 40
   30 A = PI + ATAN(Y/X)
   40 RETURN
      END
      SUBROUTINE DBUF (NORECS)
C
      COMMON /CDBUF/ LENGT,NEXT,IFIRST,IXBUF,BUFF(1024),KS,ITI
      DATA (ICOUNT=0)
      DATA (IEF=10000000000000B),(IPR=2000000000000000B)
      DATA (ITO=24012005370000000000B)
C
      IF (NORECS) 70,10,70
   10 CALL LTRIO (ITI,111B,BUFF(IFIRST),BUFF(IFIRST+511),KS)
      IF (KS .LT. 0) GO TO 76
      IF ((KS.AND.IPR) .NE. 0) PRINT 100
      IF ((KS.AND.IEF) .NE. 0) GO TO 60
      IFIRST = MOD(IFIRST+512,1024)
      ICOUNT = 0
      RETURN
   60 ICOUNT = ICOUNT+1
      IF (ICOUNT .LT. 2) GO TO 10
      PRINT 64
      CALL LTRIO (ITO,115B,A,B,JS)
      STOP
   70 IF (MOD(LENGT,512)) 79,74,79
   74 CALL LTRIO (ITI,111B,BUFF(IFIRST),BUFF(IFIRST+511),KS)
      IFIRST = MOD(IFIRST+512,1024)
   79 LENGT  = MOD(LENGT+8*NORECS,1024)
      IXBUF = NEXT
      NEXT □ LENGT +1
      RETURN
   76 PRINT 77,KS
      STOP
  100 FORMAT(/ • TROUBLE IN INPUT TAPE • /)
   64 FORMAT (*1  JOB TERMINATED--  END OF DATA*)
   77 FORMAT (*1  JOB ABORTED-- STATUS WORD   * ,O20)
      END
      SUBROUTINE OBUF (NORECS)
      COMMON /ODBUF/ LENGT,NEXT,IFIRST,IXBUF,BUFF(1024),ENDFLQ,ITO
C
      IF (NORECS) 10,40,10
   10 LENGT = LENGT + NORECS
      IF(LENGT.GT.64) GO TO 20
      IXBUF = NEXT
      GO TO 30
   20 CALL LTRIO (ITO,112B,BUFF(IFIRST),BUFF(IFIRST+511),KS)
      IF (KS .LT. 0) GO TO 80
      IXBUF = IFIRST = MOD(IFIRST+512,1024)
      LENGT = NORECS
   30 NEXT = IXBUF + NORECS*8
      RETURN
   40 INDEX = IFIRST+(LENGT-NORECS)*8-1
      CALL LTRIO (ITO,112B,BUFF(IFIRST),BUFF(INDEX),KS)
      IF (KS .LT. 0) GO TO 80
      ENDFLQ = 0.
      RETURN
   80 PRINT 10C,KS
      STOP
  100 FORMAT (*1  JOB ABORTED-- STATUS WORD   * ,O20)
      END
      PROGRAM PLTDTRD(INPUT,PUNCH,TAPE1,TAPE2,TAPE3,TAPE4,TAPE5,
     1                OUTPUT=TAPE5)
      COMMON /CDBUF/ LENGT,NEXT,JFIRST,INBUF,BUFFIN(1024),ENDFLQ,JSAP
     1 ,IT2,IT4
      COMMON /SV/ SAVE(512),ISV
      COMMON /SBUFF/ LEN(2),IFIRST(2),NETX(2),IXBUF(2),BUFT(1024,2),
     1  MODE(2)
      COMMON /RD/ R(36),IND(22)
      COMMON /DH/ TOL,ZUMX(12),ZUMY(12),SX(12),SY(12),DDX(8),DDY(8),
     1  TMTP(6,12),ISTA(12),ITN(12),NF,IJ,IK,IL,IN
      COMMON /ST/ B(13),A1,B1,C1,A2,B2,C2,DD,EE,FG,LL,RT(25),X12(25),
     1  Y12(25),LRT,NPT2,LPT2,NI(9),II(9),IE,MCT(2),COSLAT,SINLAT
      COMMON /AR/ XPS(12),YPS(12),PTN(12),D(12),NP,XPSS(12,2),YPSS(12,2)
      COMMON /GRIDP/ NR(100),NCR(100),MEA(100),MEU(100)
      COMMON /BAS/ X(1000),Y(1000),NXY(1000),XX(1000),YY(1000),VX(1000),
     1  VY(1000),X0(1000),Y0(1000),IPLT,NVENT
      DIMENSION XSAT(1000),YSAT(1000),NSAT(1000),XSAT2(1000),
     1  YSAT2(1000),NSAT2(1000)
      DIMENSION HEAD(7),XFP(8),YFP(8),APAR(6,12),BPAR(6),NREJ(100),
     1  STANAM(6),ISS(9),IDUM1(9),IDUM2(9),ICNT(12),IX1(25),IY1(25),
     2  ICD(25),NUM(8),INUM(4,2),XSTOR(8,12),YSTOR(8,12),NSTOR(8),XS(2),
     3  YS(2),ZUMA(12),XF(8,2),YF(8,2),ISM(2),PAR(6),RAP(6),OPAR(6,12),
     4  ROE(12),XC(2),YC(2),IHDS(2),KRNT(168,4),PRNT(168),NAM(150),
     5  TITLE(9),IORDER(3),IPL(2)
      EQUIVALENCE (NUM,INUM)
      DATA (IPL=1HB,1HA),(CC1=.15707963267949E-1),
     1  (C6=.17453292519943E-1),(C7=.29088820866572E-3),
     2  (C8=.48481368110953E-5),(PI=3.1415926535898),(IDUM=0),(IT=1),
     3  (ITP2=2),(ITP9=9),(IHIST=2),(W1=1.),(W2=1.),(W3=0.),
     4  (IORDER=5,0,0)
      DATA(TITLE=1H ,1H ,1H ,10H HISTOGRAM,10H OF DIFFER,10HENCES AFTE,
     1  10HR MATCHING,10H   (STARS),1H )
      DATA (IT2=24012005350000000000B),(IT4=24012005370000000000B),
     1  (IEF=10000000000000B),(ENDFLQ=0.)
      IZYXW = 0
      CALL PATCH
      IF(IZYXW.EQ.0) GO TO 100
   99 PRINT 720
      PRINT 722, (BUFFIN(I),I=1,1024)
      END FILE 5
      IF ((JSAP.AND.IEF).NE.0) GO TO 100
      CALL LTRIO (IT2,4B,BUFFIN(1),BUFFIN(1),JSAP)
  100 IZYXW = 1
      LENGT=0 $ JFIRST=NEXT=1
      IF(ENDFLQ.EQ.0.) CALL LTRIO(IT4,115B,XQ,XQ,KS)
      ENDFLQ = 1.
  101 IF (UNIT,1) 101,102,102,102
  102 REWIND 1
      MODE(1)=LEN(1)=0
      IFIRST(1)=NETX(1)=1
  103 IF (UNIT,3) 103,104,104,104
  104 REWIND 3
      MODE(2)=LEN(2)=0
      IFIRST(2)=NETX(2)=1
      ISG3=ISG4=IK=ISV=0
      IJ=IL=ISG2=1
      IPG=-1
      DO 105 I=1,12
      ICNT(I)=ZUMX(I)=ZUMY(I)=0.
      DO 105 J=1,6
      APAR(J,I) = 0.
  105 CONTINUE
      JR=MR=IU=0
      DO 1055 I=1,100
      NR(I)=NCR(I)=MEA(I)=MEU(I)=88
 1055 CONTINUE
C
C     READ HEADER CARDS
C
      CALL DBUF(0)
      CALL DBUF(1)
      DECODE (80,800,BUFFIN(INBUF)) (HEAD(I),I=1,7)
      PRINT 997
      PRINT 900,(HEAD(I),I=1,7)
      CALL DBUF(1)
      DECODE (80,799,BUFFIN(INBUF)) TOL,TLMCH,TLNUM
      NF = 0
 1062 NF = NF+1
      CALL DBUF(1)
      DECODE (80,814,BUFFIN(INBUF)) (TMTP(I,NF),I=1,4),ISTA(NF),ITN(NF),
     1  (TMTP(I,NF),I=5,6),ITEST
      IF(ITEST.EQ.0) GO TO 1062
      CALL DBUF(1)
      DECODE (80,801,BUFFIN(INBUF)) (IND(I),I=1,22),REJECT,NVENT,ICF
      IF(IND(22).LT.1) IND(22) = 0
      DO 106 I=1,8
      CALL DBUF(1)
      DECODE(80,802,BUFFIN(INBUF))XFP(I),YFP(I)
  106 CONTINUE
      NP ◘ 0
  107 NP = NP+1
      CALL DBUF(1)
      DECODE (80,803,BUFFIN(INBUF)) XPS(NP),YPS(NP),PTN(NP),ITEST
      IF(ITEST.EQ.0) GO TO 107
      IF(IND(15).EQ.0) GO TO 111
  110 CALL DBUF(1)
      DECODE (80,808,BUFFIN(INBUF)) MQ,NQ
      IF(MQ.EQ.IPL(2)) NQ = NQ+6
      CALL DBUF(1)
      DECODE (80,804,BUFFIN(INBUF)) (APAR(J,NQ),J=1,6),ITEST
      IF(ITEST.EQ.0) GO TO 110
  111 IF(IND(16).EQ.0) GO TO 112
      CALL DBUF(1)
      DECODE (80,804,BUFFIN(INBUF)) (BPAR(I),I=1,6)
  112 IR = 0
      IF(IND(22).EQ.0) GO TO 115
  113 IR = IR+1
      CALL DBUF(1)
      DECODE(80,813,BUFFIN(INBUF)) NREJ(IR),ITEST
      NR(IR) = NREJ(IR)/1000
      IF(ITEST.EQ.0) GO TO 113
C
C     READ COMPARATOR PARAMETERS
C
```

```
  115 CALL DBUF(1)
      DECODE (80,805,BUFFIN(INBUF)) JCAM,ICAM,IPLT
      CALL DBUF(1)
      DECODE (80,810,BUFFIN(INBUF)) NCOM,(R(I),I=3,6)
      DO 116 NQ=7,32,5
      NZ = NQ+4
      CALL DBUF(1)
      DECODE (80,806,BUFFIN(INBUF)) (R(I),I=NQ,NZ)
  116 CONTINUE
      D(1) = 0.
      DO 119 I=2,NP
      D(I) □ SQRT((XPS(I)-XPS(1))**2 + (YPS(I)-YPS(1))**2)
  119 CONTINUE
      CALL DBUF(1)
      DECODE (80,806,BUFFIN(INBUF)) (B(I),I=1,3)
      CALL DBUF(1)
      DECODE (80,806,BUFFIN(INBUF)) (B(I),I=4,6)
      CALL DBUF(1)
      DECODE (80,816,BUFFIN(INBUF)) (STANAM(I),I=1,4),(B(I),I=7,13),
     1  (STANAM(I),I=5,6)
      IF(B(7).LT.1.) B(7) = 0.
      CALL DBUF(1)
      DECODE(80,818,BUFFIN(INBUF)) (ISS(I),IDUM1(I),IDUM2(I),I=1,9)
      KCT = 0
      DO 120 I=1,9
      IF(ISS(I).EQ.0) GO TO 120
      KCT = KCT + 1
      ISI = ISS(I)
      II(ISI) = IDUM1(I)
      NI(ISI) = IDUM2(I)
  120 CONTINUE
C
C     PRINT HEADERS
C
      PRINT 999
      CALL DATEC (DAT)
      PRINT 901,DAT,NVENT,IPLT,NCOM
      IND(1)=IND(2)=1
      PRINT 913,(IND(I),I=1,22)
      PRINT 902
      PRINT 822,(XFP(I),YFP(I),I=1,8)
      PRINT 903
      PRINT 823,(XPS(I),YPS(I),PTN(I),I=1,NP)
      IF(IND(15).EQ.0) GO TO 123
      PRINT 904
      DO 122 I=1,12
        IPLA = IPL(1)
        IF(I.GT.6) IPLA = IPL(2)
        K = I
        IF(I.GT.6) K = I-6
        DO 121 J=1,6
          IF(APAR(J,I).NE.0.) GO TO 1215
  121     CONTINUE
        GO TO 122
 1215   PRINT 824,(APAR(J,I),J=1,6),IPLA,K
  122   CONTINUE
  123 IF(IND(16).EQ.0) GO TO 125
```

```
      PRINT 905
      PRINT 824,(BPAR(J),J=1,6)
  125 PRINT 925
      PRINT 824,(B(I),I=1,6)
      PRINT 827,(STANAM(I),I=1,4),(B(I),I=7,13),(STANAM(I),I=5,6)
      PRINT 926
      PRINT 839,(ISS(I),IDUM1(I),IDUM2(I),I=1,KCT)
      IF(IR.EQ.0) GO TO 126
      PRINT 938
      PRINT 836,(NREJ(I),I=1,IR)
  126 PRINT 960,TOL
      PRINT 961,R(36),TLMCH,TLNUM
      PRINT 962,((TMTP(J,I),J=1,4),ISTA(I),ITN(I),(TMTP(J,I),J=5,6),
     1  I=1,NF)
      IF(IND(15).EQ.0) GO TO 127
      DO 1265 I=1,12
        APAR(3,I) = APAR(3,I)*C8
        APAR(6,I) = APAR(6,I)*C8
 1265   CONTINUE
  127 IF(IND(16).EQ.0) GO TO 1275
      BPAR(3) = BPAR(3)*C8
      BPAR(6) = BPAR(6)*C8
 1275 ALPHA = CC1 * B(1)
      OMEGA = CC1 * B(2)
      CAPPA = CC1 * B(3)
      SINALP = SIN (ALPHA)
      COSALP = COS (ALPHA)
      SINOMG = SIN (OMEGA)
      COSOMG = COS (OMEGA)
      SINCOP = SIN (CAPPA)
      COSCOP = COS (CAPPA)
      A1 = - COSALP * COSCOP + SINALP * SINOMG * SINCOP
      B1 = - COSOMG * SINCOP
      C1 =   SINALP * COSCOP + COSALP * SINOMG * SINCOP
      A2 = - COSALP * SINCOP - SINALP * SINOMG * COSCOP
      B2 =   COSOMG * COSCOP
      C2 =   SINALP * SINCOP - COSALP * SINOMG * COSCOP
      DD =   SINALP * COSOMG
      EE =   SINOMG
      FG =   COSALP * COSOMG
      PINR = C6*ABS(B(8)) + C7*ABS(B(9)) + C8*ABS(B(10))
      IF(B(7).EQ.1.) PINR = -PINR
      SINLAT = SIN (PINR)
      COSLAT = COS (PINR)
      WX=R(13) $ WY=R(14) $ WXY=R(15)
      J=M=NOPT=0
      ITP = 1
C
C     READ PLATE MEASUREMENTS AND STORE ON TAPE
C
  130 CALL DBUF(1)
      DECODE(80,807,BUFFIN(INBUF)) NPLT,ITYP,NPT,XT,YT,ICODE,JTEST,ITEST
      IF(JTEST.NE.0.OR.ITEST.NE.0) GO TO 145
      KPLT = NPLT
      IF(ISG4.EQ.0.OR.ITYP.NE.0) GO TO 132
      IJ = IJ+1
      IL = IJ - IK
      J=M=ISG4=0
  132 CALL SBUF(5,ITP)
      NCT = IXBUF(ITP)
      BUFT(NCT,ITP)=XTUFT(NCT+1,ITP)=YUFT(NCT+2,ITP)=ITYP
      BUFT(NCT+3,ITP)=NPTUFT(NCT+4,ITP)=ICODE
      ICNT(IJ) = ICNT(IJ) + 1
      IF(ITYP.EQ.0.OR.ITYP.EQ.3) GO TO 133
      IF(ID.EQ.3) GO TO 142
      ID=ITYP $ NOPT=0
      GO TO 130
C
C     COUNT AND MEAN FIDUCIALS AND DRILL HOLES
C
  133 IF(NOPT.NE.0) GO TO 135
  134 NOPT = NPT
      ID = ITYP
      N=SUMX=SUMY=0
  135 IF(NPT.NE.NOPT.OR.ID.NE.ITYP) GO TO 138
      N = N+1
      IF(N.LE.25) GO TO 136
      PRINT 951
      GO TO 99
  136 IF(ITYP.EQ.0) GO TO 130
      IX1(N) = XT*1.E+6 + .001
      IY1(N) = YT*1.E+6 + .001
      ICD(N) = ICODE
      SUMX = SUMX + XT
      SUMY = SUMY + YT
      GO TO 130
  138 IF(ID.EQ.3) GO TO 142
      M = M+1
      IF(M.LE.8) GO TO 140
      PRINT 952
      GO TO 99
  140 NUM(M) = N
      GO TO 134
C
C     CHECK AND STORE DRILL HOLES
C
  142 J = J+1
      CALL DRHCHK (J,N,INUM,IX1,IY1,ICD,ISG2,ISG3,ISG4,IPG)
      TN = N
      X2 = SUMX/TN
      Y2 = SUMY/TN
      CALL REDUCE (X2,Y2,X6,Y6)
      XSTOR(J,IJ) = X6
      YSTOR(J,IJ) = Y6
      NSTOR(J) = NOPT
      IF(ITYP.EQ.3) GO TO 134
      ID = ITYP
      NOPT = 0
      GO TO 130
C
C     LAST CARD OF SET (AND ERROR 2 CARDS)
C
  145 IF(ITEST.EQ.2) GO TO 130
      IF(ITEST.NE.0) GO TO 146
      JTEST = 0
      IK = IJ
      GO TO 130
C
C     LEAST SQUARES TRANSFORMATION OF DRILL HOLES TO BEST SUBSET (PATCH)
C
  146 IK1 = IK+1
      IF(IND(15).EQ.0) GO TO 1463
      L = 6
      DO 1462 I=IK1,IJ
        L = L+1
        DO 1462 J=1,6
          APAR(J,I) = APAR(J,L)
 1462     CONTINUE
 1463 NU = 0
      DO 1464 I=1,6
        NU = NU + IND(I)
 1464   CONTINUE
      XNU = NU
      DO 161 IE=1,2
      PRINT 999
      PRINT 907,IPL(IE)
      PRINT 825,JCAM,ICAM,IPLT,IPL(IE),NCOM,(R(I),I=3,6)
      PRINT 826,(R(I),I=7,36)
      IA = 1
      IF(IE.EQ.2) IA = IK1
      IB = IK
      IF(IE.EQ.2) IB = IJ
      CALL PLTCEN (XSTOR,YSTOR,12,IA,XS,YS,IE)
      DO 1465 K=IA,IB
        ZUMA(K) = SQRT(ZUMX(K)**2+ZUMY(K)**2)
        DO 1465 L=1,8
          XSTOR(L,K) = XSTOR(L,K) - XS(IE)
          YSTOR(L,K) = YSTOR(L,K) - YS(IE)
 1465     CONTINUE
      ISM1 = IA
      IF(IA.EQ.IB) GO TO 1475
      IK2 = IA + 1
      DO 147 K=IK2,IB
        IF(ZUMA(K).LT.ZUMA(ISM1)) ISM1 = K
  147   CONTINUE
 1475 ISN = ISM1 - IA + 1
      PRINT 908,ISN
      PRINT 912
      DO 148 K=1,8
        XXX = XSTOR(K,ISM1) + XS(IE)
        YYY = YSTOR(K,ISM1) + YS(IE)
        PRINT 8301,XXX,YYY,NSTOR(K)
        X(K) = XSTOR(K,ISM1)
        Y(K) = YSTOR(K,ISM1)
        NXY(K) = NSTOR(K)
        XF(K,IE) = X(K)
        YF(K,IE) = Y(K)
  148   CONTINUE
      ISM(IE) = ISM1
      IF(IA.EQ.IB) GO TO 161
      DO 160 IL=IA,IB
```

```
        IF(IL.EQ.ISM(IE)) GO TO 160
        IN □ IL - IA + 1
        PRINT 909,IN,ISN
        INO = 0
        IF(IND(15).EQ.0) GO TO 152
      DO 151 J=1,6
        PAR(J) = APAR(J,IL)
  151   CONTINUE
      DO 1515 J=1,8
        CALL TRANSF (PAR,XSTOR(J,IL),YSTOR(J,IL))
 1515   CONTINUE
  152 DO 154 J=1,8
        DX = X(J) - XSTOR(J,IL)
        DY = Y(J) - YSTOR(J,IL)
        IF(ABS(DX).LT.R(36).AND.ABS(DY).LT.R(36)) GO TO 153
        PRINT 964,NXY(J),DX,DY
        NXY(J) = 0
        GO TO 154
  153   XX(J) □ XSTOR(J,IL)
        YY(J) = YSTOR(J,IL)
        INO = INO + 1
  154   CONTINUE
      IF(INO.GE.NU) GO TO 156
      PRINT 953,INO
      END FILE 5
      GO TO 100
  156 DO 157 I=1,6
        PAR(I) = 0.
  157   CONTINUE
      CALL LSTSQ (PAR,INO,INO,XNU,0,RO,DUMM,DUMM)
      ROE(IL) □ RO
      DO 158 J=1,8
        CALL TRANSF (PAR,XSTOR(J,IL),YSTOR(J,IL))
        XF(J,IE) = XF(J,IE) + XSTOR(J,IL)
        YF(J,IE) = YF(J,IE) + YSTOR(J,IL)
  158   CONTINUE
      DO 159 I=1,6
        OPAR(I,IL) = PAR(I)
  159   CONTINUE
  160 CONTINUE
  161 CONTINUE
      ISM1 □ ISM(1)  $  ISM2 □ ISM(2)
C
C     FIND CENTER OF PLATE FROM MEAN OF PATCHED DRILL HOLES
C
  163 DO 167 I=1,2
        TN □ IK
        IF(I.EQ.2) TN = IJ-IK
        DO 164 J=1,8
          XF(J,I) = XF(J,I)/TN
          YF(J,I) = YF(J,I)/TN
  164     CONTINUE
      CALL PLTCEN (XF,YF,2,I,XC,YC,I)
      DELX = XF(3,I) - XF(1,I)
      DELY = YF(3,I) - YF(1,I)
      CALL ANGLE (DELY,DELX,TAUL)
      SNTL = SIN(TAUL)
      CSTL = COS(TAUL)
      YCON = 1.
      F2 = (YF(2,I)-YC(I))*CSTL    - (XF(2,I)-XC(I))*SNTL
      F4 = (YF(4,I)-YC(I))*CSTL    - (XF(4,I)-XC(I))*SNTL
      IF(F4.LT.F2) YCON = -1.
      XC(I) = XC(I) + XS(I)
      YC(I) = YC(I) + YS(I)
  167 CONTINUE
      PRINT 911,IPL(1),IPL(2),XC(1),YC(1),XC(2),YC(2)
C
C     TRANSFORM PRE-IDENTIFIED SATELLITES TO SYSTEM OF -B- AND -A- SETS
C
      DO 1644 I=1,2
      PRINT 914,IPL(I)
      DO 1640 J=1,8
        X(J) = XF(J,I) + XS(I) - XC(I)
        Y(J) = YF(J,I) + YS(I) - YC(I)
        Y(J) = Y(J)*YCON
        NXY(J) = NSTOR(J)
        XX(J) = XFP(J)
        YY(J) = YFP(J)
 1640   CONTINUE
      DO 1641 J=1,5
        PAR(J) = 0.
 1641   CONTINUE
      IF(I.EQ.1) PAR(6) = PI/C8
      IF(I.EQ.2) PAR(6) = 0.
      IND(1)=IND(2)=0
      CALL LSTSQ (PAR,8,8,3.,0,DUMM,DUMM,DUMM)
      DO 1642 J=1,NP
        XPSS(J,I) = XPS(J)
        YPSS(J,I) = YPS(J)
        CALL TRANSF (PAR,XPSS(J,I),YPSS(J,I))
 1642   CONTINUE
      SINCO = SIN(PAR(6))
      COSCO = COS(PAR(6))
      XXX = PAR(6)*200./PI
      PRINT 915,PAR(4),PAR(5),XXX,SINCO,COSCO,YCON
      PRINT 903
      DO 1643 J=1,NP
        XXX = XPSS(J,I) + XC(I)
        YYY = YPSS(J,I) + YC(I)
        PRINT 823,XXX,YYY,PTN(J)
 1643   CONTINUE
 1644 CONTINUE
      CALL SBUF(0,1)
  179 IF (UNIT,1) 179,180,180,180
  180 REWIND 1
      MODE(1)=1 $ LEN(1)=0
      IFIRST(1)=NETX(1)=1
      CALL SBUF(0,1)
C
C     DECIDE WHICH SET IS TO BE THE PRIMARY SET FOR MATCHING
C
      ICT1 = 0
      DO 1840 J=1,IK
        ILT1 = ICT1 + ICNT(J)
 1840   CONTINUE
      ICT2 ◘ 0
      DO 1841 J=IK1,IJ
        ICT2 ◘ ICT2 + ICNT(J)
 1841   CONTINUE
      ICT3 ◘ ICT1 + ICT2
      MCT(1)=MCT(2)=0
      IF(ZUMA(ISM1).LT.ZUMA(ISM2)) GO TO 1842
      IE=1 $ IL=0
      IHDS(1) ◘ IPL(2)  $  IHDS(2) ◘ IPL(1)
      GO TO 185
 1842 DO 1845 J=1,ICT1
        CALL SBUF(5,1)
 1845   CONTINUE
      IE=2 $ IL=IK
      IHDS(1) ◘ IPL(1)  $  IHDS(2) = IPL(2)
C
C     PERFORM COMPARATOR REDUCTION ON STARS AND SATELLITES AND SHIFT TO
C      CENTER OF PLATE
C
  185 IL ◘ IL+1
      IC ◘ ICNT(IL)
      NOPT=ID=LL=LRT=0
      DO 1855 L=1,6
        PAR(L) ◘ OPAR(L,IL)
        RAP(L) ◘ APAR(L,IL)
 1855   CONTINUE
      DO 190 K=1,IC
        CALL SBUF(5,1)
        NCT ◘ IXBUF(1)
        X2=BUFT(NCT,1)$Y2=BUFT(NCT+1,1)$ITYP=BUFT(NCT+2,1)
        NPT=BUFT(NCT+3,1)$ICODE=BUFT(NCT+4,1)
        IF(ITYP.EQ.0.OR.ITYP.EQ.3) GO TO 190
        CALL REDUCE (X2,Y2,X6,Y6)
        X6 = X6 - XC(IE)
        Y6 = Y6 - YC(IE)
        IF(IND(15).NE.0) CALL TRANSF (RAP,X6,Y6)
C
C     PATCH STAR AND SATELLITE MEASUREMENTS, NUMBER STARS AND SATELLITES
C      AND STORE ON TAPE
C
        IF(IL.NE.ISM1.AND.IL.NE.ISM2) CALL TRANSF (PAR,X6,Y6)
        XR = X6
        YR ◘ Y6*YCON
        IF(ITYP.EQ.2.OR.ID.EQ.2. GO TO 1895
 1893   CALL SATNUM (XR,YR,NPT,INOPT,IE)
        CALL SBUF(5,2)
        NCT ◘ IXBUF(2)
        BUFT(NCT,2)=XRUFT(NCT+1,2)=YRFT(NCT+2,2)=ITYP
        BUFT(NCT+3,2)=NPT
        NCT4 = NCT+4
        ENCODE (10,819,BUFT(NCT4,2)) INOPT
        MCT(IE) = MCT(IE) + 1
        GO TO 1897
 1895   CALL STRNUM (NPT,NOPT,ITYP,ID,XR,YR)
        IF(ITYP.EQ.9) GO TO 1893
 1897   IF(K.NE.IC.OR.ITYP.NE.2) GO TO 190
```

```
          ITYP = 9
          CALL STRNUM (NPT,NOPT,ITYP,ID,XR,YR)
  190     CONTINUE
        IF(IL.NE.IK.AND.IL.NE.IJ) GO TO 185
        IF(ZUMA(ISM1).LT.ZUMA(ISM2)) GO TO 1903
        IF(IL.EQ.IJ) GO TO 1907
        IE = 2
        GO TO 185
 1903 IF(IL.EQ.IK) GO TO 1907
 1904 IF (UNIT,1) 1904,1905,1905,1905
 1905 REWIND 1
      LEN(1)=0
      IFIRST(1)=NETX(1)=1
      CALL SBUF(0,1)
      IE=1 $ IL=0
      GO TO 185
C
C     OUTPUT RAW DATA WITH STARS AND SATELLITES NUMBERED
C
 1907 IF (UNIT,1) 1907,1908,1908,1908
 1908 REWIND 1
      LEN(1)=0
      IFIRST(1)=NETX(1)=1
      CALL SBUF(0,1)
      CALL SBUF(0,2)
 1909 IF (UNIT,3) 1909,1910,1910,1910
 1910 REWIND 3
      MODE(2)=1 $ LEN(2)=0
      IFIRST(2)=NETX(2)=1
      CALL SBUF(0,2)
      LENGT = 0
      JFIRST = NEXT = 1
      CALL OBUF (1)
      ENCODE (80,944,BUFFIN(INBUF)) (HEAD(I),I=1,7)
      JCT = KCT = 0
      DO 1916 IE=1,2
      CALL OBUF(1)
      ENCODE (80,847,BUFFIN(INBUF)) ICAM,IPLT,IPL(IE),NCOM,(R(I),I=3,6)
      DO 1911 NQ1=7,32,5
        NQ6 = NQ1 + 4
        CALL OBUF(1)
        ENCODE (80,848,BUFFIN(INBUF)) (R(I),I=NQ1,NQ6)
 1911   CONTINUE
      IF(ZUMA(ISM1).GE.ZUMA(ISM2)) GO TO 1357
      IF(IE.EQ.2) GO TO 1914
      IJK = MCT(2)
      DO 1912 J=1,IJK
        CALL SBUF(5,2)
 1912   CONTINUE
      GO TO 1357
 1914 IF (UNIT,3) 1914,1915,1915,1915
 1915 REWIND 3
      LEN(2)=0
      IFIRST(2)=NETX(2)=1
      CALL SBUF(0,2)
 1357 CALL SBUF(5,1)
      JCT = JCT + 1
      NCT = IXBUF(1)
      XT=BUFT(NCT,1)$YT=BUFT(NCT+1,1)$ITYP=BUFT(NCT+2,1)
      ICODE=BUFT(NCT+4,1)
      IX = XT*1.E+6 + .001
      IY = YT*1.E+6 + .001
      IF(ITYP.NE.0.AND.ITYP.NE.3) GO TO 1359
      NPT=BUFT(NCT+3,1)
      GO TO 1360
 1359 CALL SBUF(5,2)
      NCT = IXBUF(2)
      NPT = BUFT(NCT+3,2)
      IF(ITYP.NE.2) GO TO 1360
      IF(NPT.EQ.0) GO TO 1372
 1360 KCT = KCT+1
      KRNT(KCT,1)=ITYP $ KRNT(KCT,2)=NPT $ KRNT(KCT,3)=IX
      KRNT(KCT,4)=IY $ PRNT(KCT)=BUFT(NCT+4,2)
 1361 IF(KCT.LT.168.AND.JCT.NE.ICT3) GO TO 1371
      M2 = 56
      IF(M2.GT.KCT) M2 = KCT
      PRINT 933,NVENT,IPLT
      DO 1370 I=1,M2
        IF((I+112).GT.KCT) GO TO 1363
        NM = I + 112
        GO TO 1364
 1363   IF((I+56).GT.KCT) GO TO 1367
        NM = I + 56
 1364   PRINT 840,((KRNT(K,M),M=1,4),PRNT(K),K=I,NM,56)
        GO TO 1370
 1367   PRINT 840,(KRNT(I,M),M=1,4),PRNT(I)
 1370   CONTINUE
      KCT = 0
 1371 CALL OBUF(1)
      ENCODE (80,841,BUFFIN(INBUF)) KPLT,ITYP,NPT,IX,IY,ICODE,IDUM
 1372 IF(JCT.NE.ICT1.AND.JCT.NE.ICT3) GO TO 1357
      CALL OBUF(1)
      ENCODE (80,812,BUFFIN(INBUF)) IT
 1916 CONTINUE
C
C     MATCHING SECONDARY SET TO PRIMARY SET (STARS ONLY)
C
  192 IF (UNIT,3) 192,193,193,193
  193 REWIND 3
      LEN(2)=0
      IFIRST(2)=NETX(2)=1
      CALL SBUF(0,2)
      ICT1=MCT(1) $ ICT2=MCT(2)
      IF(ZUMA(ISM1).GE.ZUMA(ISM2)) GO TO 194
      ICT1=MCT(2) $ ICT2=MCT(1)
  194 PRINT 999
      PRINT 923,IHDS(1),IHDS(2),IHDS(1)
      NU □ 0
      DO 196 I=1,6
      NU = NU + IND(I+6)
  196 CONTINUE
      XNU = NU
C
C     READ SECONDARY SET FROM STORAGE
```

```
C
      INO=IRESCK=K=KL=0
      DO 202 L=1,ICT1
      CALL SBUF(5,2)
      NCT □ IXBUF(2)
      XT=BUFT(NCT,2)$YT=BUFT(NCT+1,2)$ITYP=BUFT(NCT+2,2)
      NOPT=BUFT(NCT+3,2)
      IF(NOPT.EQ.0) GO TO 202
      IF(IR.EQ.0) GO TO 198
      DO 197 I=1,IR
        IF(NOPT.EQ.NREJ(I)) GO TO 202
  197   CONTINUE
  198 XT = -XT
      YT = -YT
      IF(IND(16).EQ.0) GO TO 199
      CALL TRANSF (BPAR,XT,YT)
  199 IF(ITYP.GT.7) GO TO 201
      K = K+1
      XX(K) = XT
      YY(K) = YT
      NXY(K) □ NOPT
      GO TO 202
  201 KL = KL+1
      XSAT2(KL) = XT
      YSAT2(KL) □ YT
      NSAT2(KL) = NOPT
  202 CONTINUE
      LL = LLL = K
      KLL = KL
C
C     READ PRIMARY SET AND LOOK FOR MATCH
C
      PRINT 924,IHDS(1)
      PRINT 945
      LINCNT = 6
      K=1 $ I=M=KL=0
  205 J □ K
  206 CALL SBUF(5,2)
      NCT = IXBUF(2)
      XT=BUFT(NCT,2)$YT=BUFT(NCT+1,2)$ITYP=BUFT(NCT+2,2)
      NOPT=BUFT(NCT+3,2)
      I = I+1
      IF(NOPT.EQ.0) GO TO 213
      IF(ITYP.GT.7) GO TO 211
  207 IF(NXY(J).EQ.NOPT) GO TO 212
      J □ J+1
      IF(J.LE.LLL) GO TO 207
      PRINT 830,XT,YT,WX,WY,WXY,ITYP,NOPT
      LINCNT = LINCNT + 1
      IF(M.LT.150) GO TO 210
      PRINT 966,IHDS(1)
      GO TO 291
  210 MF = 1000-M
      X(MF) = XT
      Y(MF) = YT
      NAM(M+1) = NOPT
      M = M+1
      IF(I-ICT2)205,215,205
  211 KL □ KL+1
      XSAT(KL) = XT
      YSAT(KL) = YT
      NSAT(KL) = NOPT
  213 IF(I-ICT2)206,215,206
  212 DELX = XX(J) - XT
      DELY = YY(J) - YT
      IF(ABS(DELX).LE.TLMCH.AND.ABS(DELY).LE.TLMCH) GO TO 214
      PRINT 955,XT,YT,WX,WY,WXY,ITYP,NOPT,DELX,DELY
      LINCNT □ LINCNT + 1
      TEMP = XX(LLL)
      XX(LLL) = XX(J)
      XX(J) = TEMP
      TEMP = YY(LLL)
      YY(LLL) = YY(J)
      YY(J) = TEMP
      TEMP = NXY(LLL)
      NXY(LLL) = NXY(J)
      NXY(J) = TEMP
      LLL □ LLL - 1
      IF(I-ICT2)205,215,205
  214 X(K) = XT
      Y(K) = YT
      TEMP = XX(K)
      XX(K) = XX(J)
      XX(J) □ TEMP
      TEMP = YY(K)
      YY(k) = YY(J)
      YY(J) = TEMP
      TEMP = NXY(K)
      NXY(K) □ NXY(J)
      NXY(J) □  TEMP
      K = K+1
      INO = INO + 1
      IF(I-ICT2)205,215,205
  215 IF(M.EQ.0) GO TO 221
C
C     LOOK FOR MEASUREMENT MATCH AND RENUMBER
C
      IHD □ 0
      DO 220 IQ=1,M
        IF(K-1.EQ.LL) GO TO 221
        MF □ 1001-IQ
        DO 217 JQ=K,LL
          DELX = XX(JQ) - X(MF)
          DELY = YY(JQ) - Y(MF)
          IF(ABS(DELX).LE.TLNUM.AND.ABS(DELY).LE.TLNUM) GO TO 218
  217     CONTINUE
        GO TO 220
  218   IF(IHD.GT.0) GO TO 219
        PRINT 946,IHDS(2),IHDS(1)
        IHD = 1
  219   PRINT 947,NXY(JQ),NAM(IQ)
        X(K) = X(MF)
        Y(K) = Y(MF)
        NXY(JQ) □ NXY(K)
```

```
          NXY(K) = NAM(IQ)
          TEMP = XX(K)
          XX(K) = XX(JQ)
          XX(JQ) = TEMP
          TEMP = YY(K)
          YY(K) = YY(JQ)
          YY(JQ) = TEMP
          K = K+1
          INO = INO + 1
  220     CONTINUE
  221 KK = KKK = K-1
      IF(INO.GE.NU) GO TO 223
      PRINT 953,INO
      GO TO 291
  223 DO 224 J=1,6
        PAR(J) = 0.
  224   CONTINUE
      CALL LSTSQ (PAR,KK,INO,XNU,6,ROES,XMEAN,YMEAN)
C
C     CHECK RESIDUAL SIZE
C
      IF(IRESCK.EQ.1) GO TO 229
      IF(REJECT.EQ.0.) REJECT = 3.
      REJX = REJECT*XMEAN
      REJY = REJECT*YMEAN
      DO 228 J=1,KK
        IF(ABS(VX(J)).LT.REJX.AND.ABS(VY(J)).LT.REJY) GO TO 228
        IF(IRESCK.EQ.1) GO TO 227
        PRINT 999
        PRINT 927,REJECT
        IRESCK = 1
  227   TELG = VX(J) * 1.E+6
        TEJG = VY(J) * 1.E+6
        PRINT 940,NXY(J),TELG,TEJG
        INO = INO - 1
        NTX = NXY(J)/1000
        IF(JR.EQ.0) GO TO 2275
        DO 2273 L1=1,JR
          IF(NTX.EQ.NCR(L1)) GO TO 2277
 2273     CONTINUE
 2275   JR = JR+1
        NCR(JR) = NTX
 2277   NXY(J) = 0
  228   CONTINUE
      IF(IRESCK.EQ.1) GO TO 223
C
C     STORE MATCHED STAR POINTS ON TAPE
C
  229 IF (UNIT,1) 229,230,230,230
  230 REWIND 1
      MODE(1)=LEN(1)=0
      IFIRST(1)=NETX(1)=1
      NS = 0
      DO 232 J=1,KK
        IF(NXY(J).EQ.0) GO TO 232
        NS = NS + 1
        XM = (X(J) + XO(J))/2.
        YM = (Y(J) + YO(J))/2.
        CALL SBUF(4,1)
        NCT = IXBUF(1)
        BUFT(NCT,1)=XMFT(NCT+1,1)=YMFT(NCT+2,1)=ITP2
        BUFT(NCT+3,1)=NXY(J)
  232   CONTINUE
      KK1 = KK+1
      IF(KK1.GT.LL) GO TO 236
      PRINT 928,IHDS(2)
      PRINT 945
      DO 234 J=KK1,LL
        CALL TRANSF (PAR,XX(J),YY(J))
        PRINT 830,XX(J),YY(J),WX,WY,WXY,ITP2,NXY(J)
  234   CONTINUE
  236 CALL HISTO (XMEAN,YMEAN,KK,INO,IHIST,TITLE)
      PRINT 996
      KKS = KK
      DO 237 J=1,KK
        XO(J) = NXY(J)
  237   CONTINUE
C
C     ROTATION AND TRANSLATION OF SATELLITE POINTS
C
      PRINT 999
      PRINT 929,IHDS(1)
      PRINT 945
      KL1 = KLL-1
      DO 238 J=1,KL1
        I = KLL-J
        DO 238 K=1,I
          IF(NSAT2(K).LT.NSAT2(K+1)) GO TO 238
          TEMP = XSAT2(K)
          XSAT2(K) = XSAT2(K+1)
          XSAT2(K+1) = TEMP
          TEMP = YSAT2(K)
          YSAT2(K) = YSAT2(K+1)
          YSAT2(K+1) = TEMP
          TEMP = NSAT2(K)
          NSAT2(K) = NSAT2(K+1)
          NSAT2(K+1) = TEMP
  238     CONTINUE
      KL1 = KL-1
      DO 239 J=1,KL1
        I = KL-J
        DO 239 K=1,I
          IF(NSAT(K).LT.NSAT(K+1)) GO TO 239
          TEMP = XSAT(K)
          XSAT(K) = XSAT(K+1)
          XSAT(K+1) = TEMP
          TEMP = YSAT(K)
          YSAT(K) = YSAT(K+1)
          YSAT(K+1) = TEMP
          TEMP = NSAT(K)
          NSAT(K) = NSAT(K+1)
          NSAT(K+1) = TEMP
  239     CONTINUE
      NB=ND=SUM=XSUM=YSUM=XSUMS=YSUMS=0
      L = 1
      DO 246 J=1,KL
        IF(IR.EQ.0) GO TO 241
        DO 240 K=1,IR
        IF(NSAT(J).EQ.NREJ(K)) GO TO 246
 240      CONTINUE
 241    IF(L.GT.KLL) GO TO 2415
        IF(NSAT(J)-NSAT2(L)) 2415,243,242
2415    PRINT 830,XSAT(J),YSAT(J),W1,W2,W3,ITP9,NSAT(J)
        GO TO 246
 242    NB = NB + 1
        XX(NB) = XSAT2(L)
        YY(NB) = YSAT2(L)
        XX(NB+750) = NSAT2(L)
        L = L+1
        GO TO 241
 243    XA=XSAT(J) $ YA=YSAT(J)
        XB=XSAT2(L) $ YB=YSAT2(L)
        L = L+1
        DELX = XB - XA
        DELY = YB - YA
        IF(ABS(DELX).LE.TLMCH.AND.ABS(DELY).LE.TLMCH) GO TO 244
        PRINT 955,XA,YA,W1,W2,W3,ITP9,NSAT(J),DELX,DELY
        NB = NB + 1
        XX(NB) = XB
        YY(NB) = YB
        XX(NB+750) = NSAT(J)
        GO TO 246
 244    CALL TRANSF (PAR,XB,YB)
        XR = (XA + XB)/2.
        YR = (YA + YB)/2.
        SUM = SUM + 1.
        DX = 2.*(XA - XR)
        DY = 2.*(YA - YR)
        XSUM = XSUM + DX
        YSUM = YSUM + DY
        ND = ND + 1
        CALL SBUF(4,1)
        NCT = IXBUF(1)
        BUFT(NCT,1)=XBUFT(NCT+1,1)=YBUFT(NCT+2,1)=ITP9
        BUFT(NCT+3,1)=NSAT(J)
        X(ND) = DX*1.E+6
        Y(ND) = DY*1.E+6
        NXY(ND) = NSAT(J)
 246    CONTINUE
C
C     PRINT DIFFERENCES AFTER SATELLITE TRANSFORMATION
C
      M1 = 1
      M2 = 54
 247  IF(M2.GT.ND) M2 = ND
      PRINT 942,NVENT,IPLT
      DO 253 J=M1,M2
        IF((J+162).GT.ND) GO TO 248
        NM = J + 162
        GO TO 251
 248    IF((J+108).GT.ND) GO TO 250
```

```
        NM = J + 108
        GO TO 251
  250   IF((J+54).GT.ND) GO TO 252
        NM = J + 54
  251   PRINT 940,(NXY(K),X(K),Y(K),K=J,NM,54)
        GO TO 253
  252   PRINT 940,NXY(J),X(J),Y(J)
  253   CONTINUE
      M1 = M1 + 216
      M2 = M2 + 216
      IF(M1.LE.ND) GO TO 247
      CALL SBUF(0,1)
  255 IF (UNIT,1) 255,256,256,256
  256 REWIND 1
      MODE(1)=1 $ LEN(1)=0
      IFIRST(1)=NETX(1)=1
      CALL SBUF(0,1)
      XM = XSUM/SUM*1.E+6
      YM = YSUM/SUM*1.E+6
      DO 258 J=1,ND
        X(J) = X(J) - XM
        Y(J) = Y(J) - YM
        XSUMS = XSUMS + X(J)**2
        YSUMS = YSUMS + Y(J)**2
  258   CONTINUE
      ERRX = SQRT(XSUMS/(4.*ND))
      ERRY = SQRT(YSUMS/(4.*ND))
      ERRXY = SQRT((XSUMS+YSUMS)/(8.*ND))
      PRINT 931,XM,YM,ERRX,ERRY,ERRXY
      ERRXY = ERRXY * 1.E-6
      IF(NB.EQ.0) GO TO 263
      PRINT 932,IHDS(2)
      DO 261 J=1,NB
        CALL TRANSF (PAR,XX(J),YY(J))
        NPPT = XX(J+750)
        PRINT 830,XX(J),YY(J),W1,W2,W3,ITP9,NPPT
  261   CONTINUE
C
C     PRINT OUTPUT AND WRITE ON TAPE
C
  263 PRINT 936
      IF(ZUMA(ISM1).LT.ZUMA(ISM2)) PRINT 965
      PRINT 9451
      LINCNT = 5
      LK = NS + ND
      J = 0 $ K = 1
      DO 277 I=1,LK
        CALL SBUF(4,1)
        NCT = IXBUF(1)
        XR=BUFT(NCT,1)$YR=BUFT(NCT+1,1)$ITYP=BUFT(NCT+2,1)
        NOPT=BUFT(NCT+3,1)
        IF(ZUMA(ISM1).LT.ZUMA(ISM2)) XR = -XR
        IF(ZUMA(ISM1).LT.ZUMA(ISM2)) YR = -YR
        CALL OBUF(1)
        IF(ITYP.GT.7) GO TO 271
        ENCODE (80,811,BUFFIN(INBUF)) XR,YR,WX,WY,WXY,ITYP,NOPT,IPLT
        PRINT 8302,XR,YR,WX,WY,WXY,ITYP,NOPT,IPLT
        IF(I.NE.NS) GO TO 265
        INF = INBUF + 6
        ENCODE (20,849,BUFFIN(INF)) NOPT,IPLT,IT
  265   TNOP = NOPT
        DO 269 L=K,KKS
          L = L
          IF(TNOP.EQ.XO(L)) GO TO 270
  269     CONTINUE
  270   XX(L) □ XR
        YY(L) = YR
        K = L+1
        NTX = NOPT/1000
        IF(MR.EQ.0) GO TO 2707
        DO 2705 L1=1,MR
          IF(NTX.EQ.MEA(L1)) GO TO 276
 2705     CONTINUE
 2707   MR = MR+1
        MEA(MR) □ NTX
        GO TO 276
  271   ENCODE (80,811,BUFFIN(INBUF)) XR,YR,W1,W2,W3,ITYP,NOPT,IPLT
        PRINT, 8302,XR,YR,W1,W2,W3,ITYP,NOPT,IPLT
        J = J+1
        X(J) = XR
        Y(J) = YR
        NXY(J) = NOPT
  276   LINCNT □ LINCNT + 1
        IF(LINCNT.LT.60) GO TO 277
        PRINT 937
        PRINT 9451
        LINCNT = 3
  277   CONTINUE
      INF = INBUF + 6
      ENCODE (20,849,BUFFIN(INF)) NOPT,IPLT,IT
      IF(IND(20).EQ.0) GO TO 285
  278 IF(XO(KKS).NE.0.) GO TO 280
      KKS = KKS-1
      GO TO 278
  280 DO 283 I=1,KKS
        IF(XO(I).EQ.0.) GO TO 283
        NPPT = XO(I)
        CALL OBUF (1)
        ENCODE (80,838,BUFFIN(INBUF)) XX(I),YY(I),VX(I),VY(I),NPPT,IPLT
  283   CONTINUE
      INF = INBUF + 6
      ENCODE (20,850,BUFFIN(INF)) NPPT,IPLT,IT
C
C     CURVE FIT, HISTOGRAM, GRID PLOT, SUMMARY SHEET
C
  285 IF(IND(19).EQ.0) GO TO 290
      IF(ICF.NE.0) IORDER(1) = ICF
      CALL CURVFT (ND,IHIST,IND(21),IORDER,ROES,ERRXY,RSX,RSY)
  290 CALL GRID(IR,JR,MR,IU)
      ROES=ROES*1.E+6 $ ERRXY=ERRXY*1.E+6 $ RSX=RSX*1.E+6 $RSY=RSY*1.E+6
      XMEAN=.5*XMEAN*1.E+6 $ YMEAN=.5*YMEAN*1.E+6
      DO 2899 I=1,IJ
        IF(I.EQ.ISM1.OR.I.EQ.ISM2) GO TO 2899
        OPAR(6,I) □ OPAR(6,I)/C8
        ROE(I) = ROE(I)*1.E+6
        OPAR(1,I)=OPAR(1,I)*1.E+6 $ OPAR(2,I)=OPAR(2,I)*1.E+6
        OPAR(4,I)=OPAR(4,I)*1.E+6 $ OPAR(5,I)=OPAR(5,I)*1.E+6
 2899   CONTINUE
      PAR(1)=PAR(1)*1.E+6 $ PAR(2)=PAR(2)*1.E+6 $ PAR(4)=PAR(4)*1.E+6
      PAR(5)=PAR(5)*1.E+6 $ PAR(3)=PAR(3)/C8 $ PAR(6)=PAR(6)/C8
      PUNCH 817,NVENT,IPLT,ROES,INO,NVENT,IPLT,ERRX,ERRY,ERRXY,ND
      CALL SHIFTQ (STANAM(1),NSTA,24)
      CALL MICARD (IPLT,NVENT,NSTA,DUM,DUM,DUM,0,0)
      PRINT 999
      PRINT 973
      IDM1 = NVENT  $  IDM2 ◘ IPLT
      DO 2909 IE=1,2
      IA ◘ 1
      IF(IE.EQ.2) IA = IK1
      IB ◘ IK
      IF(IE.EQ.2) IB ◘ IJ
      IF(IE.EQ.2) IDM1 = IDM2 = 1H
      IF(ISM(IE).NE.IA) GO TO 2901
      PRINT 968,IDM1,IDM2,NCOM,IPL(IE),IT,ZUMX(IA),ZUMY(IA),SX(IA),
     1  SY(IA)
      GO TO 2902
 2901 PRINT 968,IDM1,IDM2,NCOM,IPL(IE),IT,ZUMX(IA),ZUMY(IA),SX(IA),
     1  SY(IA),(OPAR(J,IA),J=1,2),(OPAR(J,IA),J=4,6),ROE(IA)
 2902 PRINT 998
      IF(IA.EQ.IB) GO TO 2909
      IK2 ◘ IA + 1
      DO 2904 I=IK2,IB
        IL = I - IA + 1
        IF(I.EQ.ISM(IE)) GO TO 2903
        PRINT 969,IPL(IE),IL,ZUMX(I),ZUMY(I),SX(I),SY(I),(OPAR(J,I),
     1    J=1,2),(OPAR(J,I),J=4,6),ROE(I)
        GO TO 2904
 2903   PRINT 969,IPL(IE),IL,ZUMX(I),ZUMY(I),SX(I),SY(I)
 2904   PRINT 998
 2909 CONTINUE
 2910 PRINT 974
      ISN2 = ISM2 - IK
      IF(ZUMX(ISM1).LT.ZUMX(ISM2)) GO TO 2911
      PRINT 971,.IPL(1),ISM1,(PAR(J),J=1,6),XMEAN,YMEAN,ROES
      PRINT 972,IPL(2),ISN2,ERRX,ERRY,ERRXY,ND,RSX,RSY
      GO TO 291
 2911 PRINT 971,IPL(2),ISN2,(PAR(J),J=1,6),XMEAN,YMEAN,ROES
      PRINT 972,IPL(1),ISM1,ERRX,ERRY,ERRXY,ND,RSX,RSY
  291 CALL OBUF(0)
      END FILE 5
      GO TO 100
  720 FORMAT (*1JOB STEP ABORTED*/* INPUT AREA*//)
  722 FORMAT (4X,8A10)
  799 FORMAT(F5.1,2(3X,E14.8))
  800 FORMAT(18X,A3,5A10,A9)
  801 FORMAT(22I2,F6.2,10X,A5,4X,I1)
  802 FORMAT(2E14.8)
  803 FORMAT(2E14.8,32X,F6.0,13X,I1)
  804 FORMAT(6E12.8,7X,I1)
  805 FORMAT(A1,A4,1X,A4)
  806 FORMAT(1X,E14.8,1X,E14.8,1X,E14.8,1X,E14.8,1X,E14.8)
 807 FORMAT(I4,I1,I3,2F6.6,48X,I1,9X,2I1)
 808 FORMAT(A1,1X,I1)
 810 FORMAT(11X,I1,3X,4(1X,E14.8))
 811 FORMAT(2E14.7,3E10.3,I2,I6,1X,A4)
 812 FORMAT(79X,I1)
 813 FORMAT(I6,73X,I1)
 814 FORMAT(3A10,A4,A1,1X,I1,A10,A8,24X,I1)
 816 FORMAT(A7,3A8,F1.0,1X,2F4.0,F8.4,F5.0,F4.0,F8.4,2A7)
 817 FORMAT(1X,A5,1X,A4,29X,F6.2,I4,28X,1H1/1X,A5,1X,A4,27X,3F5.1,I3,
    1  21X,2H-2)
 818 FORMAT(1X,9(I1,I2,1X,I2,2X))
 819 FORMAT(I6)
 822 FORMAT(2X,2E14.7)
 823 FORMAT(2X,2E14.7,32X,F6.0)
 824 FORMAT(1X,6E14.7,5X,A1,1H-,I1)
 825 FORMAT(1X,A1,A4,A4,A1,I3,3X,4(1X,E14.7))
 826 FORMAT(3X,E14.7,1X,E14.7,1X,E14.7,1X,E14.7,1X,E14.7)
 827 FORMAT(1X,A7,3A8,F1.0,1X,2F3.0,F8.4,F4.0,F3.0,F8.4,2A7)
 830 FORMAT(1X,2E14.7,3E10.3,I2,I6,1X,A4)
8301 FORMAT(1X,2E14.7,32X,I6)
8302 FORMAT(*T*,2E14.7,3E10.3,I2,I6,1X,A4)
 836 FORMAT(I9)
 838 FORMAT(4(1X,E14.7),1X,I6,1X,A4)
 839 FORMAT(2X,9(I1,I2,1X,I2,2X))
 840 FORMAT(*T*,3(1X,I1,1X,I6,2X,I6,2X,I6,3X,A6,10X))
 841 FORMAT(I4,I1,3I6,45X,I1,10X,I1)
 847 FORMAT(A4,A4,A1,2X,I1,3X,4(1X,E14.7))
 848 FORMAT(5(1X,E14.7))
 849 FORMAT(I6,1X,A4,8X,I1)
 850 FORMAT(1X,I6,1X,A4,7X,I1)
 900 FORMAT(1H1,30(/),60X,* JOB STEP *,A3,////52X,5A10,A9)
 901 FORMAT(*  PLATE DATA REDUCTION*,40X,* DATE OF COMPUTATION *A10//
    1  3X,11H EVENT NO. ,A5,11H PLATE NO. ,A4,5X,15H COMPARATOR NO.,I3
    2 ///)
 902 FORMAT(//*  PRELIMINARY DRILL HOLES*/)
 903 FORMAT(//27H  PRE-IDENTIFIED SATELLITES/)
 904 FORMAT(//31H  INPUT PARAMETERS FOR PATCHING/)
 905 FORMAT(//31H  INPUT PARAMETERS FOR MATCHING/)
 907 FORMAT(15X,20H ********  DATA SET ,A1,1X,9H ********///23X,14H IN
    1PUT HEADERS/)
 908 FORMAT(///16X,* SUBSET*I3,* TAKEN AS STANDARD *)
 909 FORMAT(1H1/25X,* PATCH SUBSET*I3,* TO SUBSET*I3 /)
 911 FORMAT(1H1/5X,* CENTER SHIFT FOR -*,A1,5H- SET,15X,* CENTER SHIFT
    1FOR -*,A1,5H- SET//6X,8H TRANS X,7X,8H TRANS Y,17X,8H TRANS X,7X,
    2  8H TRANS Y/2X,2(1X,E14.7),10X,2(1X,E14.7)//)
 912 FORMAT(//13H  DRILL HOLES/)
 913 FORMAT(*  PATCHING AND MATCHING SOLUTIONS SET AT *,6I2,1X,6I2//
    1  *  OTHER OPTIONS *,51X,10I2)
 914 FORMAT (1H1,*  TRANSFORMATION OF PRE-IDENTIFIED SATELLITES TO -*,
    1  A1,5H- SET)
 915 FORMAT(  //33X,* TRANSFORMATION PARAMETERS*//4X,8H TRANS X,7X,8H T
    1RANS Y,5X,11H ROT(GRADS),8X,4H SIN,11X,4H COS,10X,5H YCON/6(1X,
    2 E14.7)/)
 923 FORMAT(*  SET -* A1,*- TAKEN AS STANDARD FOR MATCHING */////7X,24H
    1 ******** MATCHING SET - A1,10H- TO SET - A1,10H- ********//)
 924 FORMAT(*  UNMATCHED -* A1,*- STAR POINTS */)
 925 FORMAT(//37H  CAMERA ORIENTATION AND STATION DATA/)
 926 FORMAT(//51H  INITIAL IMAGE AND NUMBER OF IMAGES FOR EACH TRAIL/)
 927 FORMAT(87H  SOLUTION MUST BE RECOMPUTED BECAUSE THE FOLLOWING POIN
    1TS HAVE RESIDUALS GREATER THAN ,F6.2,22H TIMES THE SQRT OF SUM//
    2 22H  PT NO    DLX      DLY)
 928 FORMAT(1H1/*  TRANSFORMED UNMATCHED -* A1,*- STAR POINTS */)
 929 FORMAT(*  UNMATCHED -* A1,*- SATELLITE POINTS */)
 931 FORMAT(//11H  SUM DX/N ,F7.3,14H     SUM DY/N ,F7.3,14H      M ERR
    1 X ,F7.3,14H      M ERR Y ,F7.3,15H      M ERR XY ,F7.3/)
 932 FORMAT(1H1/*  TRANSFORMED UNMATCHED -* A1,*- SATELLITE POINTS */)
 933 FORMAT(3HTPG,41X,43H ORIGINAL MEASUREMENTS WITH POINTS NUMBERED,
    1  20X,7H EVENT A5,9H   PLATE A4/1HT,54X,16H (-B- SET FIRST)/*T*/
    2  *T*,3(34H C  PT NO     X       Y     SAT PT,10X))
 936 FORMAT(3HTPG,13H  OUTPUT DATA/*T*)
 937 FORMAT(3HTPG,21H  OUTPUT DATA (CONT.)/*T*)
 938 FORMAT(//40H THE FOLLOWING POINTS ARE TO BE REJECTED /)
 940 FORMAT(4(1X,I6,1X,F7.3,1X,F7.3,4X))
 942 FORMAT(1H1,5X,17H SATELLITE POINTS,11X,43H DIFFERENCES AFTER TRANS
    1FORMATION (MICRONS),17X,7H EVENT A5,9H   PLATE A4//4(22H  PT NO
    2 DLX     DLY,5X))
 944 FORMAT(18HB          JOB STEP ,A3,5A10,A9)
 945 FORMAT(7X,2HLX,12X,2HLY,10X,2HWX,8X,2HWY,8X,3HWXY,6X,5HPT NO)
9451 FORMAT(1HT,6X,2HLX,12X,2HLY,10X,2HWX,8X,2HWY,8X,3HWXY,6X,5HPT NO)
 946 FORMAT(////*  THE FOLLOWING POINTS WERE MATCHED BUT NUMBERED INCOR
    1RECTLY*//20H  INCORRECT NUMBER (,A1,20H)   CORRECT NUMBER (,A1,
    2  1H))
 947 FORMAT(8X,I6,16X,I6)
 951 FORMAT(32H MORE THAN 25 READINGS PER POINT)
 952 FORMAT(30H INCORRECT NUMBER OF FIDUCIALS)
 953 FORMAT(18H NO SOLUTION ONLY  I5,15H MATCHED POINTS)
 955 FORMAT(1X,2E14.7,3E10.3,I2,I6,2X,29H COMMON POINT REJECTED  DX □
    1  E9.2,6H DY □ E9.2)
 960 FORMAT(//*  TOLERANCE FOR DRILL HOLE CHECK     TOTAL • F6.1,*
    1INDIVIDUAL   3.5*)
 961 FORMAT(//*  PRE-PATCHING TOLERANCE * E14.7///*  PRE-MATCHING TOLER
    1ANCE * E14.7,5X,* RENUMBERING TOLERANCE • E14.7)
 962 FORMAT(//*  TIME AND TEMPERATURE DATA *//(1X,3A10,A4,A1,1H-,I1,
    1  A10,A8))
 964 FORMAT(*  COMMON POINT *I6,16H REJECTED  DX □ E14.7,6H DY = E14.7)
 965 FORMAT(*T   DATA ROTATED 180 DEGREES TO -A- ORIENTATION */*T*)
 966 FORMAT(*  MORE THAN 150 UNMATCHED -* A1,*- POINTS *)
 968 FORMAT(1X,A5,2X,A4,1X,I2,3X,A1,1H-,I1,2X,2(1X,F5.1),1X,2F5.2,4X,
    1  2(1X,F5.2),1X,2(1X,F6.2),2X,F6.2,1X,F5.2)
 969 FORMAT(18X,A1,1H-,I1,2X,2(1X,F5.1),1X,2F5.2,4X,2(1X,F5.2),1X,
    1  2(1X,F6.2),2X,F6.2,1X,F5.2)
 971 FORMAT(18X,A1,1H-,I1,2X,2˙1X,F6.2),3(1X,F5.2),1X,F6.2,1X,3F5.2/)
 972 FORMAT(18X,A1,1H-,I1,59X,3F5.2,I5,2(1X,F5.2)/)
 973 FORMAT(//////////22X,28H *       C L O S U R E       *,11X,16H P A T
    1 C H I N G,11X,2H *//88H EVENT  PLT  CMP  SET     D H SHIFT   MN E
    2RROR        SCALE        DELTA      ROT    M E/26X,47H X      Y
    3 X     Y       X     Y      X     Y/)
 974 FORMAT(/////22X,2H *,19X,16H M A T C H I N G,19X,2H *,7X,20H S A T
    1 E L L I T E S,6X,2H *//27X,86H SCALE       AL      DELTA      ROT
    2    MEAN ERROR       MEAS M ERR    PTS  CRVFT M ERR/26X,85H X
    3 Y           X     Y                 X     Y   XY       X     Y   XY
    4     X     Y/)
 996 FORMAT(3HTQQ)
 997 FORMAT(1HT)
```

```
  998 FORMAT(1H )
  999 FORMAT(1H1)
      END
      SUBROUTINE REDUCE (X2,Y2,X6,Y6)
      COMMON /RD/ R(36),IND(22)
      SINA = SIN(R(19))
      X4 = X2*R(16) + Y2*R(17)*SINA
      Y4 = Y2*R(17)
      IF(IND(14))34,33,34
   33 X5=Y5=0
      GO TO 35
   34 FIX4 □ R(18)*X4
      FIY4 = R(18)*Y4
      SNX4=SIN (FIX4)
      CSX4=COS (FIX4)
      SNY4=SIN (FIY4)
      CSY4=COS (FIY4)
      SN2X4=2.*SNX4*CSX4
      CS2X4=(2.*CSX4**2)-1.
      SN2Y4= 2.* SNY4*CSY4
      CS2Y4=(2.*CSY4 **2)-1.
      SN3X4=(3.*SNX4*CSX4**2)-SNX4**3
      CS3X4=CSX4**3-(3.*SNX4**2*CSX4)
      SN3Y4=(3.*SNY4*CSY4**2)-SNY4**3
      CS3Y4=CSY4**3-(3.*SNY4**2*CSY4)
      X5 = (R(21)/2. + R(22)*CSX4 + R(23)*SNX4 + R(24)*CS2X4 + R(25)*
     1  SN2X4 + R(26)*CS3X4 + R(27)*SN3X4)
      Y5 = (R(28)/2. + R(29)*CSY4 + R(30)*SNY4 + R(31)*CS2Y4 + R(32)*
     1  SN2Y4 + R(33)*CS3Y4 + R(34)*SN3Y4)
   35 X6 = X4 + X5
      Y6 = Y4 + Y5
      RETURN
      END
      SUBROUTINE DRHCHK (J,N,INUM,IXI,IYI,ICD,ISG2,ISG3,ISG5,IPG)
      DIMENSION IXI(25),IYI(25),ICD(25),INUM(4,2),L(2),MES(2)
      COMMON /BAS/ DUM(9000),IPLT,NVENT
      COMMON /DH/ TOL,SUMX(12),SUMY(12),SEX(12),SEY(12),D1(8),D2(8),
     1  TM(6,12),ISTA(12),ITN(12),NF,IJ,IK,IL,IN
      COMMON /CDBUF/ LENGT,NEXT,JFIRST,INBUF,BUFFIN(1024),ENDFLQ,JSAP
     1 ,ITI,ITO
      COMMON /SV/ SAVE(512),ISV
      DATA MES/2H  ,2H**/
      IPLTST = 1HB
      IF(IK.GT.0) IPLTST = 1HA
      IF(J.NE.1) GO TO 9
      DO 1 M=1,8
        D1(M) = 1000.
        D2(M) = 1000.
    1   CONTINUE
      IF(IPG.LE.0) PRINT 200
      IF(IPG.EQ.1) PRINT 201
      IF(IPG.GE.0) GO TO 2
      PRINT 202
      IPG = 0
    2 DO 6 IN=1,NF
        IF(ISTA(IN).NE.IPLTST) GO TO 6
        IF(ITN(IN).EQ.IL) GO TO 7
    6   CONTINUE
      PRINT 150,IL,IPLTST
      IN = 0
      GO TO 9
    7 PRINT 151,(TM(M,IN),M=1,4),ISTA(IN),ITN(IN),(TM(M,IN),M=5,6)
    9 ISG=0 $ JK=1
      DO 10 I=2,N
        IF(ICD(I).EQ.ICD(I-1)) GO TO 10
        ISG = ISG + 1
        IP = I-1
   10   CONTINUE
      IF(ISG.EQ.1) GO TO 30
      IF(J.GT.4) GO TO 20
      NNUM = INUM(J,1) + INUM(J,2)
      IF(NNUM.NE.N) GO TO 12
      IP = INUM(J,1)
      GO TO 30
   12 JK = 2
      IF(N.LT.6) GO TO 99
      IP = N/2
      GO TO 30
   20 DO 25 I=1,2
        L(I) = 0
        IF(INUM(1,I).EQ.INUM(2,I).AND.INUM(1,I).EQ.INUM(3,I)) L(I) = 4
        IF(INUM(1,I).EQ.INUM(2,I).AND.INUM(1,I).EQ.INUM(4,I)) L(I) = 3
        IF(INUM(1,I).EQ.INUM(3,I).AND.INUM(1,I).EQ.INUM(4,I)) L(I) = 2
        IF(INUM(2,I).EQ.INUM(3,I).AND.INUM(2,I).EQ.INUM(4,I)) L(I) = 1
        IF(INUM(1,I).EQ.INUM(2,I).AND.INUM(1,I).EQ.INUM(3,I).AND.
     1    INUM(1,I).EQ.INUM(4,I)) L(I) = 5
   25   CONTINUE
      IF(L(1).EQ.5.AND.L(2).NE.0) GO TO 27
      IF(L(2).EQ.5.AND.L(1).NE.0) GO TO 27
      IF(L(1).NE.0.AND.L(1).EQ.L(2)) GO TO 27
      GO TO 29
   27 K = 1
      IF(L(1).EQ.1.OR.L(2).EQ.1) K=2
      NNUM = INUM(K,1) + INUM(K,2)
      IF(NNUM.NE.N) GO TO 12
      IP = INUM(K,1)
      GO TO 30
   29 JK = 2
      IF(N.LT.6) GO TO 99
      IP = N/2
   30 IX2=IY2=IX3=IY3=0
      DO 35 M=1,IP
        IX2 = IX2 + IXI(M)
        IY2 = IY2 + IYI(M)
   35 CONTINUE
      X2 = FLOAT(IX2)/FLOAT(IP)
      Y2 = FLOAT(IY2)/FLOAT(IP)
        IQ = IP+1
      DO 36 M=IQ,N
        IX3 = IX3 + IXI(M)
        IY3 = IY3 + IYI(M)
   36   CONTINUE
        IR = N-IP
      X3 = FLOAT(IX3)/FLOAT(IR)
      Y3 = FLOAT(IY3)/FLOAT(IR)
      D1(J) = X3 - X2
      D2(J) = Y3 - Y2
      SUMX(IJ) = SUMX(IJ) + D1(J)
      SUMY(IJ) = SUMY(IJ) + D2(J)
      IF(J.EQ.1) PRINT 79
      PRINT 80,X2,X3,Y2,Y3,D1(J),D2(J),MES(JK)
      IF(J.LT.8) RETURN
   37 ISG5 = 1
      IF(IN.NE.0) CALL MONTH (TM(3,IN),MNUM)
      IF(ISG2.LE.3) GO TO 38
      IF(IN.NE.0) PUNCH 153,(TM(M,IN),M=1,4),ISTA(IN),ITN(IN),(TM(M,IN),
     1  M=5,6),MNUM
      IF(IN.EQ.0) PUNCH 155,NVENT,IPLT,IPLTST,IL
      SUMX(IJ) = 1000. $ SUMY(IJ) = 1000.
      GO TO 44
   38 IF(ISG2-2) 39,40,41
   39 DX = SUMX(IJ)/8. $ DY = SUMY(IJ)/8.
      GO TO 42
   40 DX = SUMX(IJ)/7. $ DY = SUMY(IJ)/7.
      SUMX(IJ) = 8.*DX $ SUMY(IJ) = 8.*DY
      GO TO 42
   41 DX = SUMX(IJ)/6. $ DY = SUMY(IJ)/6.
      SUMX(IJ) = 8.*DX $ SUMY(IJ) = 8.*DY
   42 IF((ABS(SUMX(IJ))-.04).GT.TOL.OR.(ABS(SUMY(IJ))-.04).GT.TOL)
     1  ISG2 = -ISG2
      SX = SY = 0.
      DO 43 M=1,8
        IF(D1(M).EQ.1000..AND.D2(M).EQ.1000.) GO TO 43
        IF((ABS(D1(M))-.04).GT.3.5.OR.(ABS(D2(M))-.04).GT.3.5)
     1    ISG2 = -IABS(ISG2)
        D1(M) = D1(M) - DX
        D2(M) = D2(M) - DY
        SX = SX + D1(M)**2
        SY = SY + D2(M)**2
 43     CONTINUE
      SX = SQRT(SX/32.)
      SY = SQRT(SY/32.)
      IF(IABS(ISG2).NE.2) GO TO 432
      SX=SX*8./7. $ SY=SY*8./7.
      GO TO 434
432   IF(IABS(ISG2).NE.3) GO TO 434
      SX=SX*8./6. $ SY=SY*8./6.
434   PRINT 81,SUMX(IJ),SUMY(IJ),DX,DY,SX,SY
      SEX(IJ)=SX $ SEY(IJ)=SY
      IF(IN.NE.0) PUNCH 152,(TM(M,IN),M=1,4),ISTA(IN),ITN(IN),(TM(M,IN),
     1  M=5,6),SUMX(IJ),SUMY(IJ),MNUM
      IF(IN.EQ.0) PUNCH 154,NVENT,IPLT,IPLTST,IL,SUMX(IJ),SUMY(IJ)
 44   IF(ISG2.GT.0.AND.ISG2.LT.4) PRINT 956,IL,IPLTST
      IF(ISG2.LT.0) PRINT 957,IL,IPLTST
      JSG2 = 9 - IABS(ISG2)
      IF(IABS(ISG2).GT.1.AND.IABS(ISG2).LT.4) PRINT 961,JSG2
      IF(ISG2.GT.3) PRINT 958,IL,IPLTST
      ISG2=1 $ ISG3=0
 45   IPG = IPG+1
      IPG = MOD(IPG,2)
      RETURN
 99   PRINT 89,J
100   IF(ISG3.EQ.1) GO TO 101
      PRINT 998
      IF(ISV.EQ.1) PRINT 722,(SAVE(I),I=1,512)
      PRINT 997
      PRINT 722,(BUFFIN(I),I=1,512)
      PRINT 997
      PRINT 722,(BUFFIN(I),I=513,1024)
      IPG = 1
      ISG3 = 1
101   ISG2 = ISG2 + 1
      IF(J.EQ.8) GO TO 37
      RETURN
 79   FORMAT(//9X,2H X,20X,2H Y,17X,3H DX,7X,3H DY/3X,5H OPEN,5X,
     1  6H CLOSE,6X,5H OPEN,5X,6H CLOSE/)
 80   FORMAT(3(1X,F9.1,1X,F9.1,2X),A2)
 81   FORMAT(//39X,4H SUM,2X,F9.1,1X,F9.1/32X,11H BIAS ERROR,2X,F9.1,
     1  1X,F9.1/22X,21H SIGMA W/O BIAS ERROR,2X,F9.1,1X,F9.1)
 89   FORMAT(/*  DRILL HOLE *I1 * CANNOT BE CHECKED BECAUSE DRILL HOLE D
     1ATA APPEARS TO BE INCOMPLETE *)
150   FORMAT(    *  NO TIME AND TEMPERATURE DATA AVAILABLE FOR SUBSET *
     1 I2,* SET *A1)
151   FORMAT(    2X,3A10,A4,A1,1H-,I1,A10,A8)
152   FORMAT(3A10,A4,A1,1H-,I1,A10,A8,2(2X,F5.1),5X,A2)
153   FORMAT(3A10,A4,A1,1H-,I1,A10,A8,5X,1H.,6X,1H.,6X,A2)
154   FORMAT(2X,A5,3X,A4,20X,A1,1H-,I1,4X,1H-,8X,1H-,4X,2(2X,F5.1))
155   FORMAT(2X,A5,3X,A4,20X,A1,1H-,I1,4X,1H-,8X,1H-,9X,1H.,6X,1H.)
200   FORMAT(1H1)
201   FORMAT(1H //////////)
202   FORMAT(57H  NOTE..  ** INDICATES DRILL HOLE CLOSURE IS APPROXIMATE
     1D //)
722   FORMAT (4X,8A10)
956   FORMAT(///*  SUBSET*I3,5H SET A1,* MEETS TOLERANCE CHECK *)
```

```
957 FORMAT(///*  SUBSET*I3,5H SET A1,* DOES NOT MEET TOLERANCE CHECK*)
958 FORMAT(///*  SUBSET*I3,5H SET A1,* COULD NOT BE CHECKED *)
961 FORMAT(/10X,* THIS DECISION IS APPROXIMATE, MADE ON THE BASIS OF *
   1 I1,* DRILL HOLES *)
997 FORMAT(1H )
998 FORMAT(1H /1H )
    END
```

```
      SUBROUTINE MONTH (TM,MNT)
      DIMENSION ZOD(12),JOZ(13)
      DATA JOZ/2H01,2H02,2H03,2H04,2H05,2H06,2H07,2H08,2H09,2H10,2H11,
     1  2H12,2H13/
      DATA ZOD/3HJAN,3HFEB,3HMAR,3HAPR,3HMAY,3HJUN,3HJUL,3HAUG,3HSEP,
     1  3HOCT,3HNOV,3HDEC/
      DATA (YES=77777700000000000000B),(XNO=00000055555555555555B)
      TMM = TM.AND.YES.OR.XNO
      DO 10 MN=1,12
        IF(TMM.EQ.ZOD(MN)) GO TO 20
10      CONTINUE
      MN = 13
20    MNT = JOZ(MN)
      RETURN
      END
      SUBROUTINE PLTCEN (XF,YF,JZ,I,XC,YC,IZ)
      DIMENSION XF(8,JZ),YF(8,JZ),XC(2),YC(2),QQ(4,3),RN(2,3)
      K = 0
      DO 165 J=1,5,4
        K = K+1
        QQ(K,1) = YF(J+2,I) - YF(J,I)
        QQ(K,2) = XF(J,I) - XF(J+2,I)
        QQ(K,3) = XF(J,I)*YF(J+2,I) - XF(J+2,I)*YF(J,I)
        K = K+1
        QQ(K,1) = YF(J+3,I) - YF(J+1,I)
        QQ(K,2) = XF(J+1,I) - XF(J+3,I)
        QQ(K,3) = XF(J+1,I)*YF(J+3,I) - XF(J+3,I)*YF(J+1,I)
  165   CONTINUE
      DO 166 J=1,2
        DO 166 K=1,3
          RN(J,K) = 0.
          DO 166 L=1,4
            RN(J,K) = RN(J,K) + QQ(L,J)*QQ(L,K)
  166       CONTINUE
      DET = RN(1,1)*RN(2,2) - RN(1,2)*RN(2,1)
      XC(IZ) = (RN(1,3)*RN(2,2) - RN(1,2)*RN(2,3))/DET
      YC(IZ) = (RN(1,1)*RN(2,3) - RN(1,3)*RN(2,1))/DET
      RETURN
      END
      SUBROUTINE LSTSQ (PAR,KK,INO,XNU,MIND,ROE,XMEAN,YMEAN)
      COMMON /BAS/ X(1000),Y(1000),NXY(1000),XX(1000),YY(1000),VX(1000),
     1  VY(1000),X0(1000),Y0(1000),IPLT,NVENT
      COMMON /RD/ R(36),IND(22)
      DIMENSION PAR(6),CX(7),CY(7),ER(6),FF(6,10),FSTOR(6,6)
      DATA (C8=.48481368110953E-5)
      T1 = INO
      PAR(3) = PAR(3)*C8
      PAR(6) = PAR(6)*C8
      DO 83 I=1,7
        CX(I)=CY(I)=0
83      CONTINUE
      DO 84 I=1,6
        ER(I) = 0
84      CONTINUE
85    TDELS = 100.
      IF(MIND.NE.0) PRINT 943,NVENT,IPLT
      PRINT 913
      NI = 0
86    SINCO = SIN(PAR(6))
      COSCO = COS(PAR(6))
      SVVX=SVVY=DELX=DELY=0
      DO 87 I=1,60
        FF(I) = 0.
87      CONTINUE
      U3 = 1. + PAR(1)
      U4 = 1. + PAR(2)
      U5 = COSCO * PAR(3)
      U6 = SINCO * PAR(3)
      DO 96 L=1,KK
        IF(NXY(L).EQ.0) GO TO 96
        U1 = XX(L) - PAR(4)
        U2 = YY(L) - PAR(5)
        CC = U1*U3 + U2*U4*PAR(3)
        CD = U2*U4
        X0(L) =  CC*COSCO + CD*SINCO
        Y0(L) = -CC*SINCO + CD*COSCO
        DO 94 J=1,6
        MINJ = MIND + J
        IF(IND(MINJ).EQ.0) GO TO 94
          GO TO (88,89,90,91,92,93),J
88        CX(J) =  U1*COSCO
          CY(J) = -U1*SINCO
          GO TO 94
89        CX(J) =  U2*(SINCO + U5)
          CY(J) =  U2*(COSCO - U6)
          GO TO 94
90        CX(J) =  U2*U4*COSCO
          CY(J) = -U2*U4*SINCO
          GO TO 94
91        CX(J) = -U3*COSCO
          CY(J) =  U3*SINCO
          GO TO 94
92        CX(J) = -U4*(U5 + SINCO)
          CY(J) =  U4*(U6 - COSCO)
          GO TO 94
93        CX(J) =  Y0(L)
```

```
          CY(J) = -X0(L)
 94       CONTINUE
        CX(7) = X(L) - X0(L)
        CY(7) = Y(L) - Y0(L)
        VX(L) = -CX(7)
        VY(L) = -CY(7)
        DELX = DELX + VX(L)
        DELY = DELY + VY(L)
        SVVX = SVVX + VX(L)*VX(L)
        SVVY = SVVY + VY(L)*VY(L)
        DO 95 J=1,6
          DO 95 K=1,7
 95         FF(J,K) = FF(J,K) + CX(J)*CX(K) + CY(J)*CY(K)
 96         CONTINUE
       DO 97 I=1,6
        IF(FF(I,I).EQ.0.) FF(I,I) = 1.E+20
 97     CONTINUE
       CALL ERWIN (6,1,FF,IS,6,10)
       IF(IS.GT.0) GO TO 99
       PRINT 957
       RETURN
 99    DO 100 I=1,6
        DO 100 J=1,6
          FSTOR(I,J) = FF(I,J)
100       CONTINUE
       DELS = (ABS(DELX) + ABS(DELY))/T1
       PRINT 832,DELS,INO
       NI = NI+1
       IF(DELS.LT.R(35).OR.NI.GT.10) GO TO 106
       TDELS = DELS
       DO 105 I=1,6
        MINI = MIND + I
        IF(IND(MINI).EQ.0) GO TO 105
        PAR(I) = PAR(I) + FF(I,7)
105     CONTINUE
       GO TO 86
106    SVV = SVVX + SVVY
       XMEAN = SQRT (SVVX/(T1-XNU))
       YMEAN = SQRT (SVVY/(T1-XNU))
       XYMEAN = SQRT(SVV/(2.*T1-XNU))
       ROEX = XMEAN*.5
       ROEY = YMEAN*.5
       ROE = XYMEAN*.5
       DO 108 I=1,6
        MINI = MIND + I
        IF(IND(MINI).EQ.0) GO TO 108
        ER(I) = ROE * SQRT(FF(I,I))
108     CONTINUE
       PAR(3) = PAR(3)/C8
       PAR(6) = PAR(6)/C8
       PRINT 914
       PRINT 915,(PAR(I),I=1,6)
       PAR(3) = PAR(3)*C8
       PAR(6) = PAR(6)*C8
       ER(3) = ER(3)/C8
       ER(6) = ER(6)/C8
       PRINT 916
```

```
      PRINT 915,(ER(I),I=1,6)
      PRINT 917
      DELX = DELX/T1
      DELY = DELY/T1
      IF(MIND.NE.0) GO TO 110
      PRINT 826,SVV,ROE,DELX,DELY
      GO TO 111
  110 PRINT 826,SVVX,ROEX,DELX
      PRINT 826,SVVY,ROEY,DELY
      PRINT 826,SVV,ROE
  111 PRINT 918
      PRINT 826,((FSTOR(I,J),J=1,6),I=1,6)
      IF (MIND.NE.0) GO TO 117
      PRINT 919
      DO 116 I=1,KK
        IF(NXY(I).EQ.0) GO TO 116
        PRINT 833,VX(I),VY(I),NXY(I)
  116   CONTINUE
      GO TO 118
  117 DO 1170 I=1,KK
        VX(I) = VX(I) * 1.E+6
        VY(I) = VY(I) * 1.E+6
 1170   CONTINUE
      M1 = 1
      M2 = 57
 1171 IF(M2.GT.KK) M2 = KK
      PRINT 941,NVENT,IPLT
      DO 1176 J=M1,M2
        IF((J+171).GT.KK) GO TO 1172
        NM = J + 171
        GO TO 1174
 1172   IF((J+114).GT.KK) GO TO 1173
        NM = J + 114
        GO TO 1174
 1173   IF((J+57).GT.KK) GO TO 1175
        NM = J + 57
 1174   PRINT 940,(NXY(K),VX(K),VY(K),K=J,NM,57)
        GO TO 1176
 1175   PRINT 940,NXY(J),VX(J),VY(J)
 1176   CONTINUE
      M1 = M1 + 228
      M2 = M2 + 228
      IF(M1.LE.KK) GO TO 1171
      DO 1177 I=1,KK
        VX(I) = VX(I) * 1.E-6
        VY(I) = VY(I) * 1.E-6
 1177   CONTINUE
  118 RETURN
  826 FORMAT(2X,E14.7,1X,E14.7,1X,E14.7,1X,E14.7,1X,E14.7,1X,E14.7)
  832 FORMAT(4X,E14.7,6X,I4)
  833 FORMAT(1X,2(1X,E14.7),34X,I6)
  913 FORMAT(/33H  LEVEL OF RESIDUALS  NO. OF PTS.)
  914 FORMAT(/15H  OUTPUT VALUES)
  915 FORMAT(5X,9H SCALE LX,6X,9H SCALE LY,5X,11H ALPHA(SEC)/1X,3(1X,
     1  E14.7)//5X,9H TRANS LX,6X,9H TRANS LY,3X,16H ROTATION K(SEC)/
     2  1X,3(1X,E14.7))
  916 FORMAT(/24H  MEAN ERROR OF UNKNOWNS)
```

```
917 FORMAT(//15H    SQ OF DIFFS,4X,11H MEAN ERROR,4X,8H SUM D/N)
918 FORMAT (//29H  INVERSE OF NORMAL EQUATIONS)
919 FORMAT(/5X,8H DIFF LX,7X,8H DIFF LY,35X,11H COMM PT NO)
940 FORMAT(4(1X,I6,1X,F7.3,1X,F7.3,4X))
941 FORMAT(1H1,9X,12H STAR POINTS,15X,37H DIFFERENCES AFTER MATCHING (
   1MICRONS),20X,7H EVENT A5,9H   PLATE A4//4(22H  PT NO    DLX     DL
   2Y,5X))
943 FORMAT(1H1/58X,7H EVENT A5,9H   PLATE A4)
957 FORMAT(36H NO SOLUTION OBTAINABLE FROM INVERSE)
    END
```

```
SUBROUTINE TRANSF (PAR,X,Y)
DIMENSION PAR(6)
SINCO = SIN(PAR(6))
COSCO = COS(PAR(6))
U1 = X - PAR(4)
U2 = Y - PAR(5)
U3 = 1. + PAR(1)
U4 = 1. + PAR(2)
CC = U1*U3 + U2*U4*PAR(3)
CD = U2*U4
X =  CC*COSCO + CD*SINCO
Y = -CC*SINCO + CD*COSCO
RETURN
END
```

```
      SUBROUTINE STRNUM (NPT,NOPT,ITYP,ID,XT,YT)
      COMMON /ST/ B(13),A(9),LL,RT(25),X12(25),Y12(25),LRT,NPT2,LPT2,
     1  NI(9),II(9),IE,MCT(2),COSLAT,SINLAT
      COMMON /SBUFF/LEN(2),IFIRST(2),NETX(2),IXBUF(2),BUFT(1024,2),
     1  MODE(2)
      DIMENSION N(25),IPL(2)
      DATA (C5=.7272205216643E-4),(C8=.48481368110953E-5),
     1  (C9=6.2831853071796),(C10=3.1415926535898),(ITP2=2)
      DATA (IPL=1HB,1HA),(IBL=1H )
      IF(ITYP.NE.2) GO TO 50
      IF(NOPT.NE.0) GO TO 47
   46 NOPT = NPT
      ID □ ITYP
   47 IF(NOPT.NE.NPT) GO TO 50
      X = XT - B(4)
      Y □ YT - B(5)
      IF(IE.EQ.1) X □ -XT - B(4)
      IF(IE.EQ.1) Y = -YT - B(5)
      Q = X*A(3)+Y*A(6)+B(6)*A(9)
      PSI = (X*A(1)+Y*A(4)+B(6)*A(7))/Q
      ADA □ (X*A(2)+Y*A(5)+B(6)*A(8))/Q
      CALL ANGLE(ADA,PSI,ANORTH)
      ASOUTH □ ANORTH + C10
      IF(ASOUTH.GE.C9) ASOUTH □ ASOUTH - C9
      TZR □ SQRT(PSI*PSI+ADA*ADA)
      ZR = ATAN (TZR)
      Z2 □ SIN (ASOUTH)
      Z3 = COS (ASOUTH)
      Z4 = SIN (ZR)
      Z5 □ COS (ZR)
      Z6 = SINLAT * Z5 - COSLAT*Z3*Z4
      Z7 = SQRT (1.-Z6*Z6)
      CALL ANGLE (Z6,Z7,DEC)
      XX □ COSLAT * Z5 + SINLAT * Z3 *Z4
      YY = Z2 * Z4
      CALL ANGLE (YY,XX,TANGLE)
      SDEC=SIN (DEC)
      CDEC=COS (DEC)
      SINT=SIN (TANGLE)
      COST=COS (TANGLE)
      DEC □ DEC-.320*C8*COSLAT*SINT*SDEC
      TANGLE = TANGLE+(.0213*C5*COSLAT*COST)/CDEC
      LL = LL+1
      RT(LL) = -TANGLE
      X12(LL) = XT
      Y12(LL) □ YT
      IF(ABS(RT(1)-RT(LL)).GE.5.) LRT=1
      NPT2 □ NPT
      LPT2 □ MOD(NPT2,10)
      RETURN
   50 NN = NI(LPT2)
      IF(LL.EQ.NN) GO TO 49
      PRINT 959,NPT2,IPL(IE),LL,NN
      DO 48 I=1,LL
        CALL SBUF(5,2)
        NCT = IXBUF(2)
        BUFT(NCT,2)=X12(LL)UFT(NCT+1,2)=Y12(LL)UFT(NCT+2,2)=ITP2
        BUFT(NCT+3,2)=0 $ BUFT(NCT+4,2)=IBL
        MCT(IE) = MCT(IE) + 1
 48     CONTINUE
      GO TO 56
 49   IF(NN.EQ.1) GO TO 535
      DO 490 LR=1,LL
        N(LR) = LR
490     CONTINUE
      IF(LRT.EQ.0) GO TO 492
      DO 491 LR=1,LL
        IF(RT(LR).LT.-2.) GO TO 491
        RT(LR) = RT(LR) - C9
491     CONTINUE
      LRT = 0
492   NM = NN - 1
      DO 53 I=1,NM
        MM = NN-I
        DO 53 K=1,MM
          IF(RT(K)-RT(K+1))52,53,53
 52       TEMP = RT(K)
          RT(K) = RT(K+1)
          RT(K+1) = TEMP
          TEMP = X12(K)
          X12(K) = X12(K+1)
          X12(K+1) = TEMP
          TEMP = Y12(K)
          Y12(K) = Y12(K+1)
          Y12(K+1) = TEMP
          TEMP = N(K)
          N(K) = N(K+1)
          N(K+1) = TEMP
 53       CONTINUE
535   JII = II(LPT2)
      IF(NN.EQ.1) NM = 0
      JNI = JII + NM
 54   LL = 0
      DO 55 I=JII,JNI
        MPT = NPT2*100 + I
        LL = LL+1
        RT(LL) = MPT
 55     CONTINUE
      IF(NN.EQ.1) GO TO 556
      DO 555 I=1,NM
        MM = NN-I
        DO 555 K=1,MM
          IF(N(K)-N(K+1)) 555,555,553
553       TEMP=N(K)
          N(K) = N(K+1)
          N(K+1) = TEMP
          TEMP=X12(K)
          X12(K) = X12(K+1)
          X12(K+1) = TEMP
          TEMP = Y12(K)
          Y12(K) = Y12(K+1)
          Y12(K+1) = TEMP
          TEMP = RT(K)
          PT(K) = RT(K+1)
```

```
         RT(K+1) □ TEMP
555      CONTINUE
556 DO 557 LL=1,NN
       CALL SBUF(5,2)
       NCT = IXBUF(2)
       BUFT(NCT,2)=X12(LL)UFT(NCT+1,2)=Y12(LL)UFT(NCT+2,2)=ITP2
       BUFT(NCT+3,2)=RT(LL) $ BUFT(NCT+4,2)=IBL
       MCT(IE) = MCT(IE) + 1
557    CONTINUE
 56 LL = 0
    IF(ITYP.EQ.2) GO TO 46
    NOPT = 0
    ID = ITYP
    RETURN
959 FORMAT(/38H  INCORRECT NUMBER OF POINTS FOR STAR I4,7H  SET - A1,
   1  1H-,8X,I2,2H /,I2)
    END
```

```
      SUBROUTINE SATNUM (X,Y,NOPT,INOPT,IE)
      COMMON /AR/ XPS(12),YPS(12),PTN(12),D(12),N,XPSS(12,2),YPSS(12,2)
      DIMENSION A(12),G(12),H(12),XX(12),YY(12)
      DO 20 I=1,N
        XX(I) = XPSS(I,IE)
        YY(I) = YPSS(I,IE)
   20   CONTINUE
    1 DEE = SQRT((X-XX(1))**2 + (Y-YY(1))**2)
      IF(DEE)3,2,3
    2 NOPT = PTN(1)
      INOPT = NOPT*100
      RETURN
    3 SUM = 0.
      DS = 1.
      DO 50 I=1,N
        A(I) = DEE - D(I)
        IF(A(I))5,4,5
    4   NOPT = PTN(I)
        INOPT = NOPT*100
        RETURN
    5   DS = DS*A(I)
   50 CONTINUE
      DO 6 I=1,N
        G(I) = 1.
    6 CONTINUE
      I = 0
    7 I = I+1
      IF(I-N)8,8,13
    8 J = 0
    9 J = J+1
      IF(J-N)10,10,12
   10 IF(J-I)11,9,11
   11 G(I) = G(I)*(D(I)-D(J))
      GO TO 9
   12 H(I) = PTN(I)/(A(I)*G(I))
      SUM □ SUM + H(I)
      GO TO 7
   13 NOPT = DS*SUM + 0.5
      INOPT = DS*SUM*100.
      RETURN
      END
      SUBROUTINE CURVFT (LC,IHIST,IVP,IORDER,ROES,ERRXY,RSVXR,RSVYR)
      COMMON /BAS/ X(1000),Y(1000),IPN(1000),RX(1000),RY(1000),RXR(1000)
     1 ,RYR(1000),X0(1000),Y0(1000),IPLATE,IEVENT
      COMMON /CDBUF/ LENGT,NEXT,JFIRST,INBUF,BUFFIN(1024),ENDF,JSAP
     1 ,ITI,ITO
      DIMENSION V(10),C(8,8),IORDER(3),D(64),A(8,10),B(8,12)
      DIMENSION PNXY(8,12),TITLE(9)
      EQUIVALENCE (B,PNXY)
      DATA (TITLE=1H ,1H ,10H     HISTO,10HGRAM OF RE,10HSIDUALS AF,
     1  10HTER PRELIM,10HINARY CURV,10HE FIT (SAT,10HELLITE)    )
      IT = 1
      IDUM = 0
C
C     CENTER POINT DETERMINATION
C
      ICEN = (IPN(LC) + IPN(1))/2
C
C     ROTATION
C
      TAU = ATAN((Y(1) - Y(LC))/(X(1) - X(LC)))
C
C     A SOLUTION IN AN UNROTATED SYSTEM MAY BE COMPUTED BY SETTING
C       TAU = 0. AT THIS POINT OF THE PROGRAM
C
      AGL    = TAU*57.295779513082
      IAGL   = AGL
      AAGL   = IAGL
      AMIN   = (AGL - AAGL)*60.
      IAMIN  = AMIN
      AAMIN  = IAMIN
      ASEC   = (AMIN - AAMIN)*60.
      IF(ASEC .LT. 0.) ASEC = ASEC - .0005
      IF(ASEC .GT. 0.) ASEC = ASEC + .0005
      SINTAU = SIN(TAU)
      COSTAU = COS(TAU)
      DO 34 I=1, LC
         RX(I) = X(I)*COSTAU + Y(I)*SINTAU
         RY(I) = Y(I)*COSTAU - X(I)*SINTAU
   34    CONTINUE
C
C     FORMATION OF NORMALS
C
      NUM = MAXO(IORDER(1), IORDER(2), IORDER(3))
      N = NUM + 1
   35 NU1 = N + 1
      NU2 = N + 2
      DO 36 I=1, N
      DO 36 J=I, NU2
         A(I,J) = 0.
   36 CONTINUE
      V(1) = 1.
      DO 39 I=1, LC
         P = IPN(I) - ICEN
         DO 38 J=1, NUM
   38       V(J+1) = P**J
         V(NU1) = RX(I)
         V(NU2) = RY(I)
      DO 39 J=1, N
      DO 39 K=J, NU2
         A(J,K) = A(J,K) + V(J)*V(K)
   39    CONTINUE
      DO 40 J=1, N
      DO 40 K=J, N
         A(K,J) = A(J,K)
   40    CONTINUE
C
C     COMPUTATION OF INDIVIDUAL SOLUTIONS
C
      IRESCK = 0
      DO 72 I=1, 3
         IF(IORDER(I)) 41, 72, 41
   41    IN = IORDER(I)
         NU = IN + 1
      DO 37 J=1, 8
         B(J,NU+1)=B(J,NU+2)=0
   37 CONTINUE
C
C     FORMATION OF PARTICULAR NORMAL SYSTEM AND SOLUTION
C
   42    DO 43 J=1, NU
         DO 43 K=J, NU
            B(J,K) = A(J,K)
   43 CONTINUE
         DO 44 J=1, NU
            B(J,NU+1) = A(J,NU1)
            B(J,NU+2) = A(J,NU2)
   44 CONTINUE
         CALL ERWIN (NU,2,B,IS,8,12)
         IF(IS .GT. 0) GO TO 45
         PRINT 3, IPLATE, IEVENT, NUM, ICEN
         PRINT 5
      RETURN
C
C     STATISTICS COMPUTATIONS
C
   45    FREE = LC-NU
         SVX=SVY=SVXR=SVYR=0
         DO 49 J=1, LC
            P = IPN(J) - ICEN
            DO 47 K=1, IN
               V(K+1) = P**K
   47 CONTINUE
            XOUTR = B(1,NU+1)
            YOUTR = B(1,NU+2)
            DO 48 K=2, NU
               XOUTR = XOUTR + V(K)*B(K,NU+1)
               YOUTR = YOUTR + V(K)*B(K,NU+2)
   48 CONTINUE
            RXR(J) = XOUTR - RX(J)
            RYR(J) = YOUTR - RY(J)
            SVXR   = SVXR  + RXR(J)*RXR(J)
            SVYR   = SVYR  + RYR(J)*RYR(J)
   49 CONTINUE
         RSVXR = SQRT(SVXR/FREE)
         RSVYR = SQRT(SVYR/FREE)
      RSVXYR □ SQRT((SVXR+SVYR)/(2.*FREE))
C
C     PRINTED OUTPUT
C
         PRINT 6
         PRINT 3, IPLATE, IEVENT, IORDER(I), ICEN
         PRINT 7, IAGL, IAMIN, ASEC, SINTAU, COSTAU
         PRINT 8
         K = -1
         DO 50 J=1, NU
            K □ K+1
            PRINT 9, K, B(J,NU+1), B(J,NU+2)
   50       CONTINUE
         PRINT 10
      J = 0
  621 J □ J+4
      K □ J-3
      PRINT 2001
      IF(J.LE.NU) GO TO 622
      J = J-1
      IF(J.LE.NU) GO TO 624
      J = J-1
      IF(J.EQ.NU) GO TO 626
      J = J-1
      IF(J.GT.NU) GO TO 628
      PRINT 17, (L,NU,PNXY(L,NU),L=1,NU)
      GO TO 628
  622 DO 623 L=1,K
  623 PRINT 22, (L,M,PNXY(L,M),M=K,J)
      L=K+1
      PRINT 23, (L,M,PNXY(L,M),M=L,J)
      L=K+2
      PRINT 24, (L,M,PNXY(L,M),M=L,J)
      PRINT 25, J,J,PNXY(J,J)
      GO TO 621
  624 DO 625 L=1,K
  625 PRINT 18, (L,M,PNXY(L,M),M=K,J)
      L=K+1
      PRINT 19, (L,M,PNXY(L,M),M=L,J)
      PRINT 20, J,J,PNXY(J,J)
      GO TO 621
  626 DO 627 L=1,K
  627 PRINT 21, (L,M,PNXY(L,M),M=K,J)
      PRINT 26, J,J,PNXY(J,J)
  628    PRINT 12, RSVXR, RSVYR, LC, SVXR, SVYR
      SV1=ROES**2 $ SV2=ERRXY**2 $ SV3=RSVXYR**2
      SV4 = SV1 - SV2 + SV3
      PRINT 27,SV1,SV2,SV3,SV4
      PWT = 6.25E-12/SV4
      PRINT 28,PWT
      DO 51 J=1,LC
        RXR(J) = RXR(J) * 1.E+6
        RYR(J) = RYR(J) * 1.E+6
   51   CONTINUE
      M1 = 1
      M2 = 57
  511 IF(M2.GT.LC) M2 = LC
         PRINT 13,IEVENT,IPLATE
      DO 52 J=M1,M2
        IF((J+171).GT.LC) GO TO 512
        NM = J + 171
      GO TO 514
  512   IF((J+114).GT.LC) GO TO 513
        NM = J + 114
        GO TO 514
  513   IF((J+57).GT.LC) GO TO 515
        NM = J + 57
  514   PRINT 940,(IPN(K),RXR(K),RYR(K),K=J,NM,57)
        GO TO 52
  515   PRINT 940,IPN(J),RXR(J),RYR(J)
   52   CONTINUE
      M1 = M1 + 228
      M2 □ M2 + 228
      IF(M1.LE.LC) GO TO 511
C     RESIDUAL CHECK
C
         IF(IRESCK .EQ. 1) GO TO 63
         DO 56 J=1, LC
         IF(ABS(RXR(J)).LT.20..AND.ABS(RYR(J)).LT.20.) GO TO 56
            IF(IRESCK .EQ. 1) GO TO 53
            PRINT 15
            IRESCK □ 1
   53       PRINT 16, IPN(J)
            P = IPN(J) - ICEN
            DO 54 K=1, NUM
               V(K+1) □ P**K
   54 CONTINUE
            V(NU1) = RX(J)
            V(NU2) = RY(J)
            DO 55 K=1, N
            DO 55 M=K, NU2
               A(K,M) = A(K,M) - V(K)*V(M)
   55 CONTINUE
            IPN(J) □ 0
   56    CONTINUE
         IF(IRESCK .LT. 1) GO TO 62
         DO 57 J=1, N
         DO 57 K=J, N
            A(K,J) □ A(J,K)
   57 CONTINUE
   58    J = 1
   59    IF(IPN(J) .GT. 0) GO TO 61
         LC = LC-1
         DO 60 K=J, LC
            X(K)  □ X(K+1)
            Y(K)  = Y(K+1)
            RX(K) = RX(K+1)
            RY(K) = RY(K+1)
            IPN(K) = IPN(K+1)
   60       CONTINUE
      IF (J .GT. LC) GO TO 42
         GO TO 59
   61    IF(J .GE. LC) GO TO 42
```

```
       J = J+1
       GO TO 59
 62    IRESCK = 1
 63 DO 635 K=1,LC
      RXR(K) = RXR(K)*1.E-6
      RYR(K) = RYR(K)*1.E-6
635   CONTINUE
    IF(IVP.EQ.0) GO TO 66
    DO 65 K=1,LC
      CALL OBUF (1)
      ENCODE(80,2,BUFFIN(INBUF)) X(K),Y(K),RXR(K),RYR(K),IPN(K),IPLATE
 65   CONTINUE
    INF = INBUF + 6
    ENCODE (20,29,BUFFIN(INF)) IPN(LC),IPLATE,IT
 66 CALL HISTO (RSVXR,RSVYR,LC,LC,IHIST,TITLE)
    CALL FRQNCY (RSVXR,RSVYR,LC)
    PRINT 996
 72 CONTINUE
    RETURN
  2 FORMAT (4(1X,E14.7),1X,I6,1X,A4)
  3 FORMAT(39H  ADJUSTMENT OF CURVE TO SATELLITE PATH/14H  PLATE NUMBE
   1R,1X,A5,8H  EVENT A5/I3, 30H-TH ORDER POLYNOMIAL  CENTER = I4//)
  5 FORMAT (48H THE NORMAL MATRIX FOR THIS PROBLEM IS SINGULAR.)
  6 FORMAT (1H1 )
  7 FORMAT(20H  ROTATION CONSTANTS//7H  TAU □ I4, 8H DEGREES I4,
   18H MINUTES F8.3, 8H SECONDS /13H  SIN(TAU) = F10.8,
   212H COS(TAU) = F10.8//)
  8 FORMAT(27H  SOLUTION OF NORMAL SYSTEM//57H  DEGREE OF TERM   COEFF
   1ICIENTS FOR X  COEFFICIENTS FOR Y /)
  9 FORMAT (8(I15, 5X, E14.7, 6X, E14.7))
 10 FORMAT (/// 24X, 26H INVERSE OF NORMAL MATRIX )
 12 FORMAT(///33X,15H MEAN ERROR RX 9X,15H MEAN ERROR RY /33X,E14.7,
   110X, E14.7// 40X, 26H SUM OF RESIDUALS SQUARED /
   218H NUMBER OF POINTS 18X, 8H FOR RX 17X, 8H FOR RY / 7X, I4, 22X,
   3E14.7, 10X, E14.7)
 13 FORMAT(1H1,5X,17H SATELLITE POINTS,14X,36H RESIDUALS AFTER CURVE F
   1IT (MICRONS),21X,7H EVENT A5,9H   PLATE A5//4(21H  PT NO    VX
   2  VY,6X))
 15 FORMAT(//67H  PROBLEM MUST BE RECOMPUTED SINCE THE FOLLOWING POINT
   1S DO NOT HAVE/33H  RESIDUALS LESS THAN 20 MICRONS.)
 16 FORMAT (33X, I4)
 17 FORMAT(15X,      2H (,I1,1H,I1,2H) ,E14.7)
 18 FORMAT(13X,3(2X,2H (,I1,1H,I1,2H) ,E14.7))
 19 FORMAT(36X,2(2X,2H (,I1,1H,I1,2H) ,E14.7))
 20 FORMAT(61X,      2H (,I1,1H,I1,2H) ,E14.7)
 21 FORMAT(13X,2(2X,2H (,I1,1H,I1,2H) ,E14.7))
 22 FORMAT(13X,4(2X,2H (,I1,1H,I1,2H) ,E14.7))
 23 FORMAT(36X,3(2X,2H (,I1,1H,I1,2H) ,E14.7))
 24 FORMAT(59X,2(2X,2H (,I1,1H,I1,2H) ,E14.7))
 25 FORMAT(84X,      2H (,I1,1H,I1,2H) ,E14.7)
 26 FORMAT(38X,      2H (,I1,1H,I1,2H) ,E14.7)
 27 FORMAT(////*  SIGMA SQUARED*,6X,10H STAR MEAS,8X,9H SAT MEAS,
   1  7X,13H SAT CURVEFIT,15X,6H PLATE/15X,3(4X,E14.7),10X,E14.7)
 28 FORMAT(//21H  S.C.O. PLATE WEIGHT,5X,E14.7)
 29 FORMAT (1X,I6,1X,A4,7X,I1)
940 FORMAT(4(1X,I6,1X,F7.3,1X,F7.3,4X))
996 FORMAT(3HTQQ)
```

```
2001 FORMAT(1H )
     END
```

```
      SUBROUTINE HISTO (ERROR1,ERROR2,M,INO,IHIST,TITLE)
      COMMON /BAS/ DUM1(2000),IPN(1000),DUM2(2000),A(1000),B(1000),
     1  AHOLD(1000),BHOLD(1000),IPLATE,IEVENT
      DIMENSION TITLE(9),LABEL(2),GAUX(151),GAUY(151),FBLK1(32),FBLK2(32
     1),STORE(32),OX(64),OY(65),YMAX(2)
      DATA (LABEL(1)=10H X VALUES ),(LABEL(2)=10H Y VALUES ),(JUMPER=0),
     1  (ZERO=0.),(SCALE=10.)
      PRINT 996
      IF (JUMPER)140,35,140
   35 XVAL = -3.25
      DO 40 I=1,151
      GAUX(I) □ XVAL
      GAUY(I) = 3.98942280 * EXP(-XVAL*XVAL/2.0)
      XVAL □ XVAL + 3.25/75.
   40 CONTINUE
  140 DO 145 I = 1,M
      AHOLD(I) □ A(I)
      BHOLD(I) = B(I)
  145 CONTINUE
  170 DO 180 I = 1,M
      A(I) □ A(I)/ERROR1
      B(I) = B(I)/ERROR2
  180 CONTINUE
      IF (IHIST) 188,187,188
  187 N = 2
      GO TO 190
  188 N = IHIST
  190 JBLKS □ 6*N + 1
      JINTL = JBLKS
      BLK □ JBLKS
      TWON = 2*N
      BND = BLK/TWON
      BNDTL = BND
      DO 210 I = 1,JBLKS
      FBLK1(I) = 0.0
  210 FBLK2(I) = 0.0
      CRAMT = 6.5/BLK
      DO 335 I = 1,M
      IF(IPN(I).EQ.0) GO TO 335
      IF (A(I).GE.ZERO)GO TO 250
      IF ((A(I)+BNDTL).GT.ZERO)GO TO 270
      A(I) = -BNDTL
      GO TO 270
  250 IF (A(I).LT.BNDTL)GO TO 270
      A(I) = BNDTL
  270 IF (BND.LT.A(I))GO TO 310
      IF (BND.EQ.A(I))GO TO 290
      BND □ BND - CRAMT
      JBLKS = JBLKS - 1
      GO TO 270
  290 IF (A(I).LT.ZERO)GO TO 310
      FBLK1(JBLKS) = FBLK1(JBLKS) + 1.0
      GO TO 320
  310 FBLK1(JBLKS+1) = FBLK1(JBLKS+1) + 1.0
  320 BND = BNDTL
      JBLKS = JINTL
  335 CONTINUE
```

```
      DO 455 I = 1,M
      IF(IPN(I).EQ.0) GO TO 455
      IF (B(I).GE.ZERO)GO TO 370
      IF ((B(I)+BNDTL).GT.ZERO)GO TO 390
      B(I) = -BNDTL
      GO TO 390
  370 IF (B(I).LT.BNDTL)GO TO 390
  380 B(I) = BNDTL
  390 IF (BND.LT.B(I))GO TO 430
      IF (BND.EQ.B(I))GO TO 410
      BND = BND - CRAMT
      JBLKS = JBLKS - 1
      GO TO 390
  410 IF (B(I).LT.ZERO)GO TO 430
      FBLK2(JBLKS) = FBLK2(JBLKS) + 1.0
      GO TO 440
  430 FBLK2(JBLKS+1) □ FBLK2(JBLKS+1) + 1.0
  440 BND □ BNDTL
      JBLKS = JINTL
  455 CONTINUE
      SM = 1.0/M
      SN = N
      DO 480 I = 1,JINTL
      FBLK1(I) = SCALE • SN * FBLK1(I) • SM
  480 FBLK2(I) = SCALE * SN * FBLK2(I) * SM
      JPLUS = JINTL + 1
      FBLK1(JPLUS) = 0.0
      FBLK2(JPLUS) = 0.0
      DO 1155 I=1,32
        STORE(I) = FBLK1(I)
 1155   CONTINUE
      DUM = 0.
      JCT = 2*JPLUS
      XMIN = -3.50  $  YMIN □ 0.
      XMAX = 3.50
      XTCK = .01  $  YTCK □ .01
      DX = .5  $  DY = 1.
      XLAB = 4.1
      YLAB = 2.0
      JSIG = 0
      XGRD □ 3.5
      DO 1230 I=1,2
      YMAX(I) = 4.
      DO 1200 L=1,JPLUS
      IF(YMAX(I).LT.STORE(L)) YMAX(I) = STORE(L)
 1200 CONTINUE
      YMAX(I) = YMAX(I) + .25
      XVAL = -3.25
      OY(1) = 0.
      DO 1220 L=1,JPLUS
        CK = L-1
        OX(2*L-1) = XVAL + CK*CRAMT
        OX(2*L) = OX(2*L-1)
        OY(2*L) = STORE(L)
        OY(2*L+1) = OY(2*L)
 1220   CONTINUE
      ISIG = 0
      PRINT 15,ISIG,JSIG,XMIN,YMIN,XMAX,YMAX(I),XGRD,YMAX(I),XTCK,YTCK,
     1  DX,DY,XLAB,YLAB,JCT
      PRINT 18, TITLE,LABEL(I)
      PRINT 16, (OX(J),J=1,JCT)
      PRINT 16, (OY(J),J=1,JCT)
      PRINT 17
      ISIG ◘ 1
      KG ◘ 151
      PRINT 15,ISIG,JSIG,DUM,DUM,DUM,DUM,DUM,DUM,DUM,DUM,DUM,DUM,DUM,
     1  DUM,KG
      PRINT 16, ( GAUX(J),J=1,151)
      PRINT 16, ( GAUY(J),J=1,151)
      PRINT 17
      DO 1225 K=1,32
        STORE(K) = FBLKZ(K)
 1225   CONTINUE
 1230 CONTINUE
 1235 DO 1238 I = 1,M
      A(I) = AHOLD(I)
 1238 B(I) = BHOLD(I)
      JUMPER = 1
 1280 RETURN
C     FORMAT SPECIFICATIONS
   15 FORMAT(1HT,2I1,12E10.3,I4)
   16 FORMAT(1HT,12E11.4)
   17 FORMAT(*TT*)
   18 FORMAT(1HT,10A10)
  996 FORMAT(3HTQQ)
      END
      SUBROUTINE GRID (IR,JR,MR,IU)
C
C     THIS VERSION OF GRID IS FOR PRODUCTION PLATES.
C
      DIMENSION IX(4), TITLE(20,4)
      DIMENSION IP(100,4),AL(10,10),NU(10,10),OT(10,10),OO(10),OP(10)
C
      COMMON /BAS/ DUM(9000),IPLT,NVENT
      COMMON /GRIDP/ IP
C
      DATA (DOT=1H.), (DOTS=7H.......), (DOTI=7H......I),
     1     (ION=1HI), (BLKI=7H      I), (BLKD=7H      .),
     1     (BLK=1H ), (BLKS=7H       )
      DATA (IRD=1)
C
      IX(1)=IR $ IX(2)=JR $ IX(3)=MR
      IF(IRD.EQ.0) GO TO 10
      READ 1,((NU(I,J),J=1,10),I=1,10)
      READ 2,((AL(I,J),J=1,10),I=1,10)
C
   10 IU = 0
      DO 50 I=1,10
      IF(I.GT.1.AND.I.LT.10) GO TO 11
      J1=4 $ J2=7
      GO TO 15
   11 IF(I.GT.2.AND.I.LT.9) GO TO 12
      J1=3 $ J2=8
      GO TO 15
   12 IF(I.GT.3.AND.I.LT.8) GO TO 13
      J1=2 $ J2=9
      GO TO 15
   13 J1=1 $ J2=10
   15 DO 40 J=J1,J2
      DO 30 K=1,3
      L1 = IX(K)
      IF(L1.EQ.0) GO TO 30
      DO 25 L=1,L1
      IF(IP(L,K).EQ.NU(I,J)) GO TO 40
   25 CONTINUE
   30 CONTINUE
      IU = IU+1
      IP(IU,4) = NU(I,J)
   40 CONTINUE
   50 CONTINUE
      IX(4) = IU
C
      DO 500 K=1,4
      IF(IRD.EQ.0) GO TO 60
      READ 3,(TITLE(L1,K),L1=1,20)
   60 PRINT 4,NVENT,IPLT,(TITLE(L1,K),L1=1,20)
C
      DO 100 I=1,10
      DO 70 J=1,10
      OT(I,J) = AL(I,J)
   70 CONTINUE
      IF(I.GT.1.AND.I.LT.10) GO TO 71
      J1=4 $ J2=7
      GO TO 75
   71 IF(I.GT.2.AND.I.LT.9) GO TO 72
      J1=3 $ J2=8
      GO TO 75
   72 IF(I.GT.3.AND.I.LT.8) GO TO 73
      J1=2 $ J2=9
      GO TO 75
   73 J1=1 $ J2=10
   75 DO 90 J=J1,J2
      L2 = IX(K)
      IF(L2.EQ.0) GO TO 82
      DO 80 L=1,L2
      IF(IP(L,K).EQ.NU(I,J)) GO TO 90
   80 CONTINUE
   82 OT(I,J) = BLKI
   90 CONTINUE
  100 CONTINUE
C
      DO 105 I=1,10
      OO(I) = BLKS
      IF(I.EQ.3) OO(I) = BLKD
      IF(I.GT.3.AND.I.LT.8) OO(I) = DOTS
  105 CONTINUE
      PRINT 5,BLK,(OO(J),J=1,10)
C
      DO 110 I=1,10
      OP(I) = BLKS
      IF(I.GT.2.AND.I.LT.8) OP(I) = BLKI
  110 CONTINUE
      DO 120 I=3,7
  120 OO(I) = DOTI
      OO(2)=BLKD $ OO(8)=DOTS
      PRINT 5,BLK,(OP(J),J=1,10)
      PRINT 5,BLK,(OP(J),J=1,10)
      PRINT 5,BLK,(OT(1,J),J=1,10)
      PRINT 5,BLK,(OP(J),J=1,10)
      PRINT 5,BLK,(OO(J),J=1,10)
C
      OP(2)=OP(8)=BLKI
      OO(1)=BLKD $ OO(2)=OO(8)=DOTI $ OO(9)=DOTS
      PRINT 5,BLK,(OP(J),J=1,10)
      PRINT 5,BLK,(OP(J),J=1,10)
      PRINT 5,BLK,(OT(2,J),J=1,10)
      PRINT 5,BLK,(OP(J),J=1,10)
      PRINT 5,BLK,(OO(J),J=1,10)
C
      OP(1)=OP(9)=BLKI
      OO(1)=OO(9)=DOTI $ OO(10)=DOTS
      PRINT 5,BLK,(OP(J),J=1,10)
      PRINT 5,BLK,(OP(J),J=1,10)
      PRINT 5,BLK,(OT(3,J),J=1,10)
      PRINT 5,BLK,(OP(J),J=1,10)
      PRINT 5,DOT,(OO(J),J=1,10)
C
      OP(10)=BLKI $ OO(10)=DOTI
      DO 200 I=4,7
      PRINT 5,ION,(OP(J),J=1,10)
      PRINT 5,ION,(OP(J),J=1,10)
      PRINT 5,ION,(OT(I,J),J=1,10)
      PRINT 5,ION,(OP(J),J=1,10)
      PRINT 5,ION,(OO(J),J=1,10)
  200 CONTINUE
C
      OP(10)=OO(10)=BLKS $ OO(1)=BLKI
      PRINT 5,BLK,(OP(J),J=1,10)
      PRINT 5,BLK,(OP(J),J=1,10)
      PRINT 5,BLK,(OT(8,J),J=1,10)
      PRINT 5,BLK,(OP(J),J=1,10)
      PRINT 5,BLK,(OO(J),J=1,10)
C
      OP(1)=OO(1)=OP(9)=OO(9)=BLKS $ OO(2)=BLKI
      PRINT 5,BLK,(OP(J),J=1,10)
      PRINT 5,BLK,(OP(J),J=1,10)
      PRINT 5,BLK,(OT(9,J),J=1,10)
      PRINT 5,BLK,(OP(J),J=1,10)
      PRINT 5,BLK,(OO(J),J=1,10)
C
      OP(2)=OO(2)=OP(8)=OO(8)=BLKS $ OO(3)=BLKI
      PRINT 5,BLK,(OP(J),J=1,10)
      PRINT 5,BLK,(OP(J),J=1,10)
      PRINT 5,BLK,(OT(10,J),J=1,10)
      PRINT 5,BLK,(OP(J),J=1,10)
      PRINT 5,BLK,(OO(J),J=1,10)
  500 CONTINUE
      IRD = 0
      RETURN
C
    1 FORMAT(10(2X,I2,3X))
    2 FORMAT(10A7)
    3 FORMAT(10A8)
    4 FORMAT(3HTPG,88X,7H EVENT A5,9H   PLATE A5/*T*/*T*,27X,10A8/*T*/
     1  *T*,27X,10A8/*T*/*T*/*T*/*T*)
    5 FORMAT(*T*,31X,A1,10A7)
      END
      SUBROUTINE DBUF (NORECS)
C
      COMMON /CDBUF/ LENGT ,NEXT,IFIRST,IXBUF,BUFF(1024),ENDFLQ,KS
     1 ,ITI,ITO
      COMMON /SV/ SAVE(512),ISV
      DATA (ICOUNT=0)
      DATA (IEF=10000000000000B),(IPR=2000000000000000B)
C
      IF (NORECS) 70,10,70
   10 CALL LTRIO (ITI,111B,BUFF(IFIRST),BUFF(IFIRST+511),KS)
      IF (KS .LT. 0) GO TO 76
      IF ((KS.AND.IPR) .NE. 0) PRINT 100
      IF ((KS.AND.IEF) .NE. 0) GO TO 60
      IFIRST = MOD(IFIRST+512,1024)
      ICOUNT = 0
      RETURN
   60 ICOUNT = ICOUNT+1
      IF (ICOUNT .LT. 2) GO TO 10
      PRINT 64
      END FILE 5
      END FILE 5
      CALL LTRIO (ITO,115B,A,B,JS)
      STOP
   70 IF(MOD(LENGT,512)) 79,74,79
   74 DO 75 I=1,512
      IFF = IFIRST+I-1
      SAVE(I) □ BUFF(IFF)
   75 CONTINUE
      ISV = 1
      CALL LTRIO (ITI,111B,BUFF(IFIRST),BUFF(IFIRST+511),KS)
      IFIRST = MOD(IFIRST+512,1024)
   79 LENGT = MOD(LENGT+8*NORECS,1024)
      IXBUF = NEXT
      NEXT = LENGT +1
      RETURN
   76 PRINT 77,KS
      STOP
  100 FORMAT(/ * TROUBLE IN INPUT TAPE * /)
   64 FORMAT (*1  JOB TERMINATED--  END OF DATA*)
   77 FORMAT (*1  JOB ABORTED-- STATUS WORD   * ,O20)
      END
      SUBROUTINE MICARD (NPLT,NVNT,NSTA,PLT,VNT,STA,ISIG,JSIG)
      IF(ISIG.EQ.1) GO TO 50
      NAM1 = NPLT.AND.77000000000000000000B
      NAM2 = NPLT.AND.770000000000000000B
      CALL SHIFTQ(NAM2,MAN2,6)
      NAM3 = NPLT.AND.7777000000000000B
      CALL SHIFTQ(NAM3,MAN3,12)
      CALL SHIFTQ(NAM2,MNA,12)
      IF(MNA.LE.32B) GO TO 10
      PUNCH 1,NVNT,NAM1,MAN2,MAN3,NSTA
      IF(JSIG.EQ.1) PRINT 1,NVNT,NAM1,MAN2,MAN3,NSTA
      IF(JSIG.EQ.2) PRINT 3,NVNT,NAM1,MAN2,MAN3,NSTA
      GO TO 100
   10 PUNCH 2,NVNT,NAM1,MAN2,MAN3,NSTA
      IF(JSIG.EQ.1) PRINT 2,NVNT,NAM1,MAN2,MAN3,NSTA
      IF(JSIG.EQ.2) PRINT 4,NVNT,NAM1,MAN2,MAN3,NSTA
      GO TO 100
   50 ABC1 = PLT.AND.77000000000000000000B
      ABC2 = PLT.AND.770000000000000000B
      CALL SHIFTQ(ABC2,CBA2,6)
      ABC3 = PLT.AND.7777000000000000B
      CALL SHIFTQ(ABC3,CBA3,12)
      CALL SHIFTQ(ABC2,MNA,12)
      IF(MNA.LE.32B) GO TO 60
      PUNCH 1,VNT,ABC1,CBA2,CBA3,STA
      IF(JSIG.EQ.1) PRINT 1,VNT,ABC1,CBA2,CBA3,STA
      IF(JSIG.EQ.2) PRINT 3,VNT,ABC1,CBA2,CBA3,STA
      GO TO 100
   60 PUNCH 2,VNT,ABC1,CBA2,CBA3,STA
      IF(JSIG.EQ.1) PRINT 2,VNT,ABC1,CBA2,CBA3,STA
      IF(JSIG.EQ.2) PRINT 4,VNT,ABC1,CBA2,CBA3,STA
  100 RETURN
    1 FORMAT(2X,A5,1X,A1,1X,A1,A2,55X,A3)
    2 FORMAT(2X,A5,1X,2A1,1X,A2,55X,A3)
    3 FORMAT(3X,A5,1X,A1,1X,A1,A2,55X,A3)
    4 FORMAT(3X,A5,1X,2A1,1X,A2,55X,A3)
      END
      SUBROUTINE FRQNCY (RSVX,RSVY,L)
      DIMENSION TITLE(9),LABEL(2),XG(2),YG(2)
      COMMON /BAS/ DUM1(1000),XPN(1000),IPN(1000),DUM2(2000),A(1000,2),
     1  DUM3(2000),IPLATE,IEVENT
      DATA (TITLE=1H ,1H ,10H FREQUENCY,10H PLOT OF R,10HESIDUALS A,
     1  10HFTER PRELI,10HMINARY CUR,10HVE FIT (SA,10HTELLITE)  )
      DATA (LABEL=10H X VALUES ,10H Y VALUES )
      DATA (N2=2),(DUM=0.)
      INC = 100
      L1 = L-1
      DO 10 I=1,L1
      NX = IPN(I+1) - IPN(I)
      IF(NX.LT.INC) INC = NX
   10 CONTINUE
      TINC = INC
      TOP = 15.
      YMIN □ -15.
      YMAX = 15.
      XGRD □ 200.*TINC +1.
      YGRD □ 15.
      XTCK = .01  $  YTCK □ .01
      XLAB □ 7.0  $  YLAB □ 4.0
      DX □ 40*INC  $  DY = 3.
      RVX □ RSVX*1.E+6  $  RVY = RSVY*1.E+6
      DO 15 I=1,L
      XPN(I) □ IPN(I)
      DO 15 J=1,2
      A(I,J) = A(I,J)*1.E+6
      IF(ABS(A(I,J)).GT.TOP) A(I,J) = SIGN(TOP,A(I,J))
   15 CONTINUE
      JSIG □ 0
      RV = RVX
      DO 65 I=1,2
      XMIN □ IPN(1) -200*INC -1
      XMAX = IPN(1)
      M = 1
   20 XMIN □ XMIN + 200.*TINC
      XMAX = XMAX + 200.*TINC
      IIN = XMIN +1.
      IAX = XMAX -1.
      ISIG □ 0
      DO 22 JJ=M,L
      IF(IPN(JJ).GE.IIN) GO TO 23
   22 CONTINUE
      GO TO 70
   23 DO 25 LL=JJ,L
      KK = JJ+L-LL
      IF(IPN(KK).LE.IAX) GO TO 26
   25 CONTINUE
      GO TO 70
   26 KK1 = KK-1
   30 DO 35 J=JJ,KK1
      IF((IPN(J+1)-IPN(J)).NE.INC) GO TO 40
   35 CONTINUE
      NPTS = KK-JJ+1
      LST = KK
      IF(NPTS.EQ.1) GO TO 50
   GO TO 43
40 NPTS = J-JJ+1
   LST □ J
   IF(NPTS.EQ.1) GO TO 50
43 PRINT 1,ISIG,JSIG,XMIN,YMIN,XMAX,YMAX,XGRD,YGRD,XTCK,YTCK,DX,DY,
  1  XLAB,YLAB,NPTS
   IF(ISIG.EQ.0) PRINT 2,TITLE,LABEL(I)
   PRINT 3,(XPN(K),K=JJ,LST)
   PRINT 3,(A(K,I),K=JJ,LST)
   PRINT 4
   ISIG = 1
45 IF(LST.EQ.KK) GO TO 46
   JJ = LST+1
   IF(JJ.LT.KK) GO TO 30
   LST = KK
   GO TO 50
46 DO 48 J=1,2
   SIDE = (-1.)**(J+1)
   XME □ SIDE*RV
   PRINT 1,ISIG,JSIG,DUM,DUM,DUM,DUM,DUM,DUM,DUM,DUM,DUM,DUM,DUM,
  1DUM,N2
   PRINT 3, XMIN,XMAX
   PRINT 3, XME,XME
   PRINT 4
48 CONTINUE
   IF (KK.EQ.L) GO TO 60
   M = KK+1
   IF(M.LT.L) GO TO 20
   GO TO 60
50 XG(1) = XPN(JJ) - .5*TINC
   XG(2) = XPN(JJ) + .5*TINC
   YG(1)=YG(2)=A(JJ,I)
   PRINT 1,ISIG,JSIG,XMIN,YMIN,XMAX,YMAX,XGRD,YGRD,XTCK,YTCK,DX,DY,
  1  XLAB,YLAB,N2
   IF(ISIG.EQ.0) PRINT 2,TITLE,LABEL(I)
   PRINT 3,XG(1),XG(2)
   PRINT 3,YG(1),YG(2)
   PRINT 4
   ISIG = 1
   GO TO 45
60 RV = RVY
65 CONTINUE
   RETURN
70 PRINT 6
   RETURN
 1 FORMAT(1HT,2I1,12E10.3,I4)
 2 FORMAT(1HT,10A10)
 3 FORMAT(1HT,12E11.4)
 4 FORMAT(2HTT)
 6 FORMAT(1X,* RESIDUALS OUT OF ORDER *)
   END
```

```
      SUBROUTINE ERWIN (N,NC,A,IS,NJ,NK)
      DIMENSION A(NJ,NK)
C     MODIFIED CHOLESKY
      IS = 1
        I = 1
      DO 10 J=1,N
        GO TO (5,3,1),I
    1   K = J-2
      DO 2 L=1,K
      DO 2 M=J,N
    2   A(J-1,M) = A(J-1,M) - A(L,J-1)*A(L,M)
    3   K = J-1
      DO 4 L=1,K
    4   A(J,J) = A(J,J) - A(L,J)*A(L,J)
        I = 2
    5 IF(A(J,J)) 6, 6, 7
    6 IS = -1
      GO TO 22
    7    A(J,N+3) = SQRT (A(J,J))
      DO 8 L=J,N
    8   A(J,L) = A(J,L)/A(J,N+3)
      DO 9 L=1,J
    9   A(L,J) = A(L,J)/A(J,N+3)
   10   I □ I+1
C     INVERSION OF U
      DO 11 I=2,N
        J = I-1
      DO 11 K=1,J
   11   A(I,K) = -A(K,I,
      DO 12 I=3,N
        J = I-2
      DO 12 K=1,J
        L □ I-K-1
        M = I
      DO 12 IJ=1,K
        M = M-1
   12   A(I,L) □ A(I,L) - A(I,M)*A(L,M)
C     COMPUTATION OF U INVERSE * U INVERSE TRANSPOSE
      DO 14 I=1,N
      DO 14 J=I,N
        A(I,N+4) □ 0.
      DO 13 K=J,N
   13   A(I,N+4) □ A(I,N+4) + A(K,I)*A(K,J)
   14   A(I,J) = A(I,N+4)
      DO 15 I=1,N
      DO 15 J=I,N
   15   A(I,J) = A(I,J)/A(I,N+3)
      DO 16 I=1,N
      DO 16 J=1,I
   16   A(J,I) □ A(J,I)/A(I,N+3)
      DO 17 K=1,N
      DO 17 L=K,N
   17   A(L,K) □ A(K,L)
C     COMPUTATION OF SOLUTIONS
      IF(NC) 18,22,18
   18 M  = N+NC
      I  = N+1
    DO 21 J=I,M
    DO 19 K=1,N
19    A(K,N+4) = 0.
    DO 20 K=1,N
    DO 20 L=1,N
20    A(K,N+4) = A(K,N+4) + A(K,L)*A(L,J)
    DO 21 K=1,N
21    A(K,J) = A(K,N+4)
22  RETURN
    END
```

```
      SUBROUTINE ANGLE (Y,X,A)
      PI = 3.1415926535898
      IF(X) 30,20,10
   10 IF(Y.LT.0.) GO TO 13
      A = ATAN(Y/X)
      GO TO 40
   13 A = 2.*PI + ATAN(Y/X)
      GO TO 40
   20 IF (Y) 23,22,21
   21 A □ PI/2.
      GO TO 40
   22 A = 0.
      GO TO 40
   23 A = 3.*PI/2.
      GO TO 40
   30 A = PI + ATAN(Y/X)
   40 RETURN
      END
      SUBROUTINE SBUF (NWRDS,NS)
C
      COMMON /SBUFF/ LEN(2),NEXT(2),IFIRST(2),IXBUF(2),BUFSCR(1024,2),
     1  MODE(2)
C
      IF (NS.EQ.1) ITJ = 1
      IF (NS.EQ.2) ITJ = 3
      IF(MODE(NS)) 105,5,105
    5 IF (NWRDS) 10,20,10
   10 LEN(NS) = LEN(NS) + NWRDS
      IF (LEN(NS).GT.512) GO TO 30
      IXBUF(NS) = NEXT(NS)
      GO TO 50
   20 IF (LEN(NS).EQ.0) RETURN
   30 IF (UNIT,ITJ) 30,35,300,300
   35 DO 700 K=1,2
      IFF = IFIRST(NS)
   38 BUFFER OUT(ITJ,1) (BUFSCR(IFF,NS),BUFSCR(IFF+511,NS))
      IF (NWRDS) 45,40,45
   40 IF (UNIT,ITJ) 40,700,300,300
  700 CONTINUE
   45 IFP = IFIRST(NS)+512
      IXBUF(NS) = IFIRST(NS) = MOD(IFP,1024)
      LEN(NS) = NWRDS
   50 NEXT(NS) = IXBUF(NS) + NWRDS
      RETURN
  105 IF (NWRDS) 170,110,170
  110 IF (UNIT,ITJ) 110,120,300,300
  120 IFF = IFIRST(NS)
      BUFFER IN(ITJ,1)(BUFSCR(IFF,NS),BUFSCR(IFF+511,NS))
  125 IF (UNIT,ITJ) 125,126,300,300
  126 IFP = IFIRST(NS)+512
      IFIRST(NS) = MOD(IFP,1024)
      RETURN
  170 IF(MOD(LEN(NS),512)) 176,175,176
  175 IF (UNIT,ITJ) 175,178,300,300
  176 LEN(NS) = LEN(NS) + NWRDS
      IF (LEN(NS).GT.512) GO TO 175
      IXBUF(NS) = NEXT(NS)
      GO TO 179
  300 PRINT 301,NS
  301 FORMAT (//* TROUBLE IN SBUF *,I2)
      STOP
  178 IFF = IFIRST(NS)
      BUFFER IN (ITJ,1) (BUFSCR (IFF,NS),BUFSCR(IFF+511,NS))
      IFP = IFIRST(NS) + 512
      IXBUF(NS) = IFIRST(NS) = MOD(IFP,1024)
      LEN(NS) = NWRDS
  179 NEXT(NS) = IXBUF(NS) + NWRDS
      RETURN
      END
      SUBROUTINE OBUF (NORECS)
      COMMON /CDBUF/ LENGT,NEXT,IFIRST,IXBUF,BUFF(1024),ENDFLQ,JSAP,
     1  ITI,ITO
C
      IF (NORECS) 10,40,10
   10 LENGT = LENGT + NORECS
      IF(LENGT.GT.64) GO TO 20
      IXBUF = NEXT
      GO TO 30
   20 CALL LTRIO (ITO,112B,BUFF(IFIRST),BUFF(IFIRST+511),KS)
      IF (KS .LT. 0) GO TO 80
      IXBUF = IFIRST = MOD(IFIRST+512,1024)
      LENGT = NORECS
   30 NEXT = IXBUF + NORECS*8
      RETURN
   40 INDEX = IFIRST+(LENGT-NORECS)*8-1
      CALL LTRIO (ITO,1;2B,BUFF(IFIRST),BUFF(INDEX),KS)
      IF (KS .LT. 0) GO TO 80
      ENDFLQ = 0.
      RETURN
   80 PRINT 100,KS
      STOP
  100 FORMAT (*1  JOB ABORTED-- STATUS WORD   * ,O20)
      END
      PROGRAM STARID (TAPE2,TAPE4,TAPE5,OUTPUT=TAPE5,JFILE,TAPE1=JFILE,
     1  SAO,TAPE3=SAC)
C
C  CAPACITY 90 TRAILS, 1000 POINTS, 200 STARS
C
      DIMENSION             QJAY(525),CATNO(68),RTASC(68),PMRA(68),
     1  DECL(68),PMDC(68),SRCBK(68),STARS(408),ASTRA(68,6),RTSV(299),
     2  HEAD(6),R(27),XTRL(90),DAYNUM(90,8),DATA(1000,7),ISG(90),
     3  COORD(200,9),TEM(204),IID(10),NID(10),IBD(4),FWT(3,2)
      COMMON /CDBUF/ LEN,NEXT,IFIRST,INBUF ,BUFFIN(1024),ENDFLO,IST
     1 ,IT2,IT4
      COMMON /CHK/ COORD,DTOL,PR1DIF,PR2DIF,STORE(2,11),FWT,INDWT
      COMMON /UPD/ DTDIF,IDIFF,IDCR,JJ,XTRL,DAYNUM,NARC,QJAY
      COMMON /TST/ NID,IID,LNCNT
      EQUIVALENCE (STARS,ASTRA,CATNO),(STARS(69),RTASC),
     1  (STARS(137),PMRA),(STARS(205),DECL),(STARS(273),PMDC),
     2  (STARS(341),SRCBK)
      DATA(PI=3.1415926535898),(TWPI=6.2831853071796),
     1 (HFPI=1.5707963267949),(PO=760.),(TO=0.),(RDG=.17453292519943E-1)
     2,(RMIN=.29088820866572E-3),(RSEC=.48481368110953E-5),
     3 (RHR=.26179938779915),(RTMIN=.43633231299858E-2),
     4 (RTSEC=.7272205216643E-4),(GRTORD=.15707963267949E-1),
     5 (AP0=.29137566E-3),(AP1=-.3227865E-6),(AP2=.10225E-8),
     6 (CM=.2234945E-3),(CN=.97169024E-4),(PMN=.21716742E-7),
     7 (DMDT=.13526302E-8),(DNDT=-.41209163E-9),(IT=1),(BLANK=5H     )
      DATA (IT2=24012005350000000000B),(IT4=24012005370000000000B),
     1  (IEF=10000000000000B),(ENDFLQ=0.)
C
C
C     READ AND STORE J-CORRECTION TABLE
      J2 = 0
      DO 1000 I=1,52
        BUFFER IN (1,1) (QJAY(1),QJAY(525))
 1002   IF (UNIT,1) 1002,1003,1003,1003
 1003   IA = (I-1)*525 + 121750
        CALL WRITEC (QJAY,IA,525,J2)
 1000   CONTINUE
C
C     READ AND STORE STAR CATALOG
      DO 1001 I=1,298
        BUFFER IN (3,1) (STARS(1),STARS(408))
 1004   IF (UNIT,3) 1004,1005,1005,1005
 1005   RTSV(I) □ RTASC(68)
        IA = (I-1)*408
        CALL WRITEC (STARS,IA,408,J2)
 1001   CONTINUE
      I = 299
      BUFFER IN (3,1) (STARS(1),STARS(162))
 1006 IF (UNIT,3) 1006,1007,1007,1007
 1007 RTSV(I) = RTASC(27)
      CALL WRITEC (STARS,121584,162,J2)
C
C     INITIALIZE ABORT ROUTINE
      IZYX = 0
      CALL PATCH
      IF(IZYX.EQ.0) GO TO 100
      PRINT 98
      PRINT 97,(BUFFIN(I),I=1,1024)
      END FILE 5
      IF((IST.AND.IEF) .NE. 0) GO TO 100
      CALL LTRIO (IT2,4B,XQ,XQ,IST)
C
C     BEGINNING OF PROGRAM PROPER
  100 IZYX = 1
      N=NN=ISORT=0
      IDCR □ 1
      DO 1012 I=1,10
        IID(I) □ NID(I) = 0
 1012   CONTINUE
      DO 1013 I=1,90
        ISG(I) □ 0
 1013   CONTINUE
      DO 101 I=1,800
        COORD(I) = 0.
  101   CONTINUE
      IF(ENDFLQ.EQ.0.) CALL LTRIO (IT4,115B,XQ,XQ,KS)
      ENDFLQ = 1.
      LEN=0 $ IFIRST=NEXT=1
      CALL DBUF(0)
      CALL DBUF(1)
      DECODE (80,5,BUFFIN(INBUF)) ISTEP,(HEAD(I),I=1,6)
      PRINT 96
      PRINT 6,ISTEP,(HEAD(I),I=1,6)
      CALL DBUF(1)
      DECODE (80,2,BUFFIN(INBUF)) RTOL,DTOL,EVENT,INDWT
      IF(INDWT.LT.1) INDWT □ 0
      IF(INDWT.EQ.0) GO TO 102
      CALL DBUF(1)
      DECODE (80,12,BUFFIN(INBUF)) (FWT(I,1),I=1,3)
      CALL DBUF(1)
      DECODE (80,12,BUFFIN(INBUF)) (FWT(I,2),I=1,3)
C
C     READ APPROXIMATE CAMERA ORIENTATION PARAMETERS
  102 CALL DBUF(1)
      DECODE (80,   3 ,BUFFIN(INBUF))(R(I),I=22,24)
      CALL DBUF(1)
      DECODE (80,   3 ,BUFFIN(INBUF))(R(I),I=25,27)
C
C     READ DATE, LOCATE IN J-CORRECTION TABLE
      CALL DBUF(1)
      DECODE (80,   4 ,BUFFIN(INBUF))ITRL   ,IPLT,IYEAR,MONTH,IDAY,IAZ
      PRINT 30,EVENT,IPLT,IYEAR,MONTH,IDAY
      PRINT 22,RTOL,DTOL,EVENT,INDWT
      IF(INDWT.NE.0) PRINT 26,((FWT(I,J),I=1,3),J=1,2)
      PRINT 23,(R(I),I=22,27)
      IF(MONTH.GT.6) GO TO 115
      NARC = 2*(IYEAR-62) - 1
      IF(MONTH.NE.1) GO TO 104
      IDATE = IDAY
      GO TO 126
  104 IF(MONTH.NE.2) GO TO 106
      IDATE = 31 + IDAY
      GO TO 126
  106 IF(MOD(IYEAR,4).NE.0) GO TO 108
```

```
      IDAY = IDAY + 1
  108 IF(MONTH.NE.3) GO TO 110
      IDATE = 59 + IDAY
      GO TO 126
  110 IF(MONTH.NE.4) GO TO 112
      IDATE □ 90 + IDAY
      GO TO 126
  112 IF(MONTH.NE.5) GO TO 114
      IDATE = 120 + IDAY
      GO TO 126
  114 IDATE = 151 + IDAY
      GO TO 126
  115 NARC = 2*(IYEAR-62)
      IF(MONTH.NE.7) GO TO 117
      IDATE = IDAY
      GO TO 126
  117 IF(MONTH.NE.8) GO TO 119
      IDATE = 31 + IDAY
      GO TO 126
  119 IF(MONTH.NE.9) GO TO 121
      IDATE □ 62 + IDAY
      GO TO 126
  121 IF(MONTH.NE.10) GO TO 123
      IDATE = 92 + IDAY
      GO TO 126
  123 IF(MONTH.NE.11) GO TO 125
      IDATE □ 123 + IDAY
      GO TO 126
  125 IDATE □ 153 + IDAY
  126 IA = (NARC-1)*525 + 121750
      CALL READEC (QJAY,IA,525,J2)
      NDATE □ QJAY(1)
      NX = NDATE/100
      NDAY □ NDATE - NX*100
      NY = NX/100
      NMNTH = IABS(NX - NY*100)
      IF(NMNTH.NE.1) GO TO 128
      NDATE = NDAY
      GO TO 129
  128 NDATE = NDAY - 30
  129 NDIFF □ IDATE - NDATE
      IDIFF = NDIFF/10
      DTDIF □ FLOAT(NDIFF)/10. - FLOAT(IDIFF)
      IDIFF = IDIFF + 1
      IF(IAZ.EQ.0) GO TO 132
C
C     COMPUTE ALPHA OMEGA KAPPA IF CAMERA ANGLES GIVEN IN AZ, ZEN DIST
      AZIMR □ R(22) * RDG
      ZENIR = R(23) * RDG
      COPAR = R(24) * RDG
      AK = AZIMR - PI
      IF(AK.LT.0.) AK □ AK + TWPI
      SINOMG = SIN(ZENIR)*SIN(AK)
      COSOMG = SQRT(1.- SINOMG**2)
      SINALP □ SINOMG/COSOMG/TAN(AK)
      COSALP □ COS(ZENIR)/COSOMG
      TANALP = SINALP /COSALP
      CALL ANGLE (TANALP,SINOMG,DELKAP)
      COPPAR = COPAR - DELKAP + HFPI
      SINCOP =SIN (COPPAR)
      COSCOP =COS (COPPAR)
      GO TO 133
  132 ALPHAR □ R(22) * GRTORD
      OMEGAR = R(23) * GRTORD
      COPPAR □ R(24) * GRTORD
      SINALP = SIN (ALPHAR)
      COSALP = COS (ALPHAR)
      SINOMG = SIN (OMEGAR)
      COSOMG = COS (OMEGAR)
      SINCOP = SIN (COPPAR)
      COSCOP = COS (COPPAR)
C
C     COMPUTE (APPROXIMATE) DIRECTION MATRIX OF CAMERA
  133 A1 = -COSALP*COSCOP + SINALP*SINOMG*SINCOP
      B1 □ -COSOMG*SINCOP
      C1 = SINALP*COSCOP+COSALP*SINOMG*SINCOP
      D □ SINALP * COSOMG
      E = SINOMG
      F = COSALP • COSOMG
      A2 = -COSALP * SINCOP - SINALP*SINOMG*COSCOP
      B2 □ COSOMG * COSCOP
      C2 □ SINALP • SINCOP - COSALP*SINOMG*COSCOP
      DPAR = R(27) * D
      EPAR = R(27) * E
      FPAR = R(27) * F
      J = 1
C
C     READ PARAMETERS FOR GIVEN TRAIL, STORE IN ARRAY DAYNUM
  134 CALL DBUF(1)
      DECODE (80,   3 ,BUFFIN(INBUF))(R(I),I=1,3)
      CALL DBUF(1)
      CALL DBUF(1)
      DECODE (80,   3 ,BUFFIN(INBUF))(R(I),I=4,6)
      CALL DBUF(1)
      DO 135 K=7,19,3
        L = K+2
        CALL DBUF(1)
        DECODE (80,3,BUFFIN(INBUF)) (R(I),I=K,L)
  135   CONTINUE
      IF(MOD(J,7).EQ.0) PRINT 99
      PRINT 24,ITRL    ,IPLT,IYEAR,MONTH,IDAY
      PRINT 25,(R(I),!=1,21)
      DTSQ = R(18)**2/200.
      PHIR □ R(1)*RDG + R(2)*RMIN + R(3)*RSEC
      SINPHI =SIN (PHIR)
      COSPHI = COS (PHIR)
      PABAR = R(20) * (1.-.00264*COS(2.*PHIR) - 2.*R(19)/6370000.)
      REFRAC □ PABAR • (1.+.003665*TO) / PO / (1.+.003665*R(21))
      XTRL(J) □ ITRL
      DAYNUM(J,1) = R(4)*RHR + R(5)*RTMIN + R(6)*RTSEC
      DAYNUM(J,2) □ R(7)*RHR + R(8)*RTMIN + R(9)*RTSEC
      DAYNUM(J,3) =R(10)*RHR +R(11)*RTMIN +R(12)*RTSEC
      DAYNUM(J,4) = R(13)*RTSEC
      DAYNUM(J,5) = R(14)*RSEC
      DAYNUM(J,6) = R(15)*RSEC
      DAYNUM(J,7) = R(16)*RSEC
      DAYNUM(J,8) = R(17)
      J = J+1
C
C     READ NEXT INPUT, TEST IF TRAIL CARD OR PLATE POINT CARD
  137 CALL DBUF(1)
      DECODE (80,   7 ,BUFFIN(INBUF))X,Y,W1,W2,W3,ITYP,NPT,ITRL,
     1 NPLT,ITEST
      IF(ITYP.NE.0) GO TO 138
      IF(J.LE.90) GO TO 134
      PRINT 90
      GO TO 142
  138 IF(ISORT.EQ.1) GO TO 141
C
C     SORT TRAIL PARAMETERS ON INCREASING TRAIL NUMBER
      JJ = J-1
      KK □ JJ-1
      DO 140 I=1,KK
        L = JJ - I
        DO 140 K=1,L
          IF(XTRL(K).LE.XTRL(K+1)) GO TO 140
          TEMP = XTRL(K)
          XTRL(K) = XTRL(K+1)
          XTRL(K+1) = TEMP
          DO 139 M=1,8
            TEMP = DAYNUM(K,M)
            DAYNUM(K,M) = DAYNUM(K+1,M)
            DAYNUM(K+1,M) = TEMP
  139       CONTINUE
  140     CONTINUE
      ISORT = 1
  141 IF(IPLT.EQ.NPLT) GO TO 143
      PRINT 91,IPLT,NPLT
  142 CALL DBUF(1)
      DECODE (80,  10 ,BUFFIN(INBUF))ITEST
      IF (ITEST) 300,142,300
C
C     STORE INFORMATION FOR PLATE POINTS IN ARRAY DATA
  143 NPTN = NPT*1000 + ITRL
      NN □ NN + 1
      IF(NN.LE.1000) GO TO 144
      PRINT 93
      GO TO 142
  144 DATA(NN,1) = X
      DATA(NN,2) = Y
      DATA(NN,3) □ W1
      DATA(NN,4) = W2
      DATA(NN,5) = W3
      DATA(NN,6) = NPTN
      DATA(NN,7) = ITRL
      IF(ITEST)145,137,145
  145 PRINT 46
C
C     SORT PLATE POINT INFORMATION ON INCREASING POINT NUMBER
      N1 = NN - 1
      DO 148 I=1,N1
```

```
    L = NN - I
    DO 148 K=1,L
     IF(DATA(K,6).LE.DATA(K+1,6)) GO TO 148
     DO 147 M=1,7
      TEMP □ DATA(K,M)
      DATA(K,M) = DATA(K+1,M)
      DATA  (K+1,M) = TEMP
 147  CONTINUE
 148 CONTINUE
   DO 165 K=1,NN
    ITRPN □ DATA(K,6)/100.
    TRPN = ITRPN
C
C  FOR EACH PLATE POINT, COMPUTE APPROXIMATE AZIMUTH AND ZEN DIST
    XL □ DATA(K,1) - R(25)
    YL □ DATA(K,2) - R(26)
    U □ XL*A1 + YL*A2 + DPAR
    V = XL*B1 + YL*B2 + EPAR
    WW= XL*C1 + YL*C2 + FPAR
    SCSI □ U/WW
    SCNU = V/WW
    CALL ANGLE (SCNU,SCSI,CAY)
    A □ CAY + PI
    TANZR = SQRT(SCSI**2 + SCNU**2)
    ZR = ATAN(TANZR)
    RM □ AP0*TANZR + AP1*TANZR**3 + AP2*TANZR**5
    Z □ ZR + RM*REFRAC
    SINZ = SIN(Z)
    COSZ = COS(Z)
    SINA = SIN(A)
    COSA □ COS(A)
C
C  COMPUTE APPROXIMATE HOUR ANGLE AND DECLINATION
    SINDEC = COSZ*SINPHI - SINZ*COSA*COSPHI
    COSCOS □ COSZ*COSPHI + SINZ*COSA*SINPHI
    COSSIN = SINZ * SINA
    CALL ANGLE (COSSIN,COSCOS,HP)
    COSDEC = SQRT(1.- SINDEC**2)
    TANDEC = SINDEC/COSDEC
    DEC = ATAN(TANDEC)
C
C  LOCATE PARAMETERS FOR TRAIL TO WHICH PLATE POINT BELONGS
    DO 152 I=1,JJ
     IF(DATA(K,7).EQ.XTRL(I)) GO TO 154
 152 CONTINUE
    PRINT 92,DATA(K,7)
    IF(TRPN.NE.COORD(N,5)) GO TO 165
C
C  IF THERE ARE PARAMETERS FOR ONLY PART OF A SET OF POINTS,
C  INDICATE BY A CODE IN COORD(N,6)
    BP = 0.
    TRLC = AINT(COORD(N,6)/10000.)
    IF(TRLC.EQ.0.) GO TO 153
    BP = 1.
    TRLC = AINT(TRLC*.1)
    IF(TRLC.EQ.0.) GO TO 153
    BP = 2.
```

```
      TRLC = AINT(TRLC*.1)
      IF(TRLC.NE.0.) BP = 3.
  153 BPP = COORD(N,4) + 1. + BP
      MB = BP
      COORD(N,6) = COORD(N,6) + BPP*10.**(MB+4)
      GO TO 165
C
C     STORE SIDEREAL TIME, COMPUTE RIGHT ASCENSION (APPROXIMATE)
  154 ISG(I) = 1
      DATA(K,7) = DAYNUM(I,1)
      RA = DAYNUM(I,1) - HP
      IF(RA.LT.0.) RA = RA + TWPI
      IF(RA.GT.TWPI) RA = RA - TWPI
C
C     BACKDATE APPROXIMATE RIGHT ASCENSION AND DECLINATION TO 1950
      RAG = RA + DAYNUM(I,2)
      RAH = RA + DAYNUM(I,3)
      SINRAG = SIN(RAG)
      COSRAG = COS(RAG)
      SINRAH = SIN(RAH)
      COSRAH = COS(RAH)
      RAO = RA - (DAYNUM(I,4) + DAYNUM(I,5)*SINRAG*TANDEC +
     1  DAYNUM(I,6)*SINRAH/COSDEC)
      DCO = DEC - (DAYNUM(I,5)*COSRAG + DAYNUM(I,6)*COSRAH*SINDEC +
     1  DAYNUM(I,7)*COSDEC)
      SINRAO = SIN(RAO)
      COSRAO = COS(RAO)
      TADECO = TAN(DCO)
      AVRT = CM + CN*SINRAO*TADECO
      AVDC = CN*COSRAO
      SVRT = 100. * (CN**2*SINRAO*COSRAO*(1.+ 2.*TADECO**2) + PMN*
     1  COSRAO*TADECO + DMDT + DNDT*SINRAO*TADECO)
      SVDC = 100. * (-CN**2*SINRAO**2* TADECO - PMN*SINRAO + DNDT*
     1  COSRAO)
      RT1950 = RAO - (AVRT*R(18) + SVRT*DTSQ)
      DE1950 = DCO - (AVDC*R(18) + SVDC*DTSQ)
      IF(RT1950.LT.0.) RT1950 = RT1950 + TWPI
C
C     STORE APPROXIMATE STAR COORDINATES IN ARRAY COORD, ACCUMULATING
C     MULTI OBSERVATIONS OF THE SAME STAR
      IF(N.EQ.0) GO TO 158
      IF(TRPN.EQ.COORD(N,5)) GO TO 160
  158 N = N+1
      IF(N.LE.200) GO TO 159
      PRINT 94
      GO TO 300
  159 COORD(N,5) = TRPN
      COORD(N,6) = K
  160 COORD(N,1) = COORD(N,1) + RT1950
      COORD(N,2) = COORD(N,2) + DE1950
      COORD(N,3) = COORD(N,3) + DEC
      COORD(N,4) = COORD(N,4) + 1.
  165 CONTINUE
C
C     MEAN STAR COORDINATE APPROXIMATIONS
      DO 168 I=1,N
      DO 168 K=1,3
```

```
          COORD(I,K) □ COORD(I,K)/COORD(I,4)
  168     CONTINUE
C
C     SORT STAR INFORMATION ON INCREASING RIGHT ASCENSION
      KK = N-1
      DO 172 I=1,KK
        L = N - I
        DO 172 M=1,L
          IF(COORD(M,1).LE.COORD(M+1,1)) GO TO 172
          DO 170 K=1,6
            TEMP □ COORD(M,K)
            COORD(M,K) □ COORD(M+1,K)
            COORD(M+1,K) = TEMP
  170       CONTINUE
  172     CONTINUE
C
C     LIST APPROXIMATE STAR COORDINATES
      M1 = 1
      M2 = 58
  173 IF(M2.GT.N) M2 = N
      PRINT 47,EVENT,IPLT
      DO 180 I=M1,M2
        IF((I+116).GT.N) GO TO 174
        NM = I + 116
        GO TO 176
  174   IF((I+58).GT.N) GO TO 178
        NM □ I + 58
  176   PRINT 48,(COORD(K,1),COORD(K,2),COORD(K,5),COORD(K,4),K=I,NM,58)
        GO TO 180
  178   PRINT 48,COORD(I,1),COORD(I,2),COORD(I,5),COORD(I,4)
  180   CONTINUE
      M1 = M1 + 174
      M2 = M2 + 174
      IF(M1.LE.N) GO TO 173
C
C     BEGIN STAR LOOKUP IN SAO CATALOG
      PRINT 34,EVENT,IPLT
      LNCNT □ 3
      RTOL = RTOL * RSEC
      DTOL = DTOL * RSEC
      CALL READEC (STARS,0,408,J2)
      JN □ 1
      JL = 1
      DO 240 I=1,N
        IF(COORD(I,1).LE.(TWPI+RTOL)) GO TO 181
        PRINT 87,COORD(I,5)
        LNCNT = LNCNT + 2
        COORD(I,3) = BLANK
        L □ AMOD(COORD(I,5),10.)
        IF(L.EQ.0) L = 10
        NID(L) = NID(L) + 1
        GO TO 239
C
C     FOR EACH STAR APPROXIMATION, SET LIMITS
  181   PR1DIF = 10.
        PR2DIF = 10.
        RALO = COORD(I,1) - RTOL
        RAHI = COORD(I,1) + RTOL
        IF(RALO.GE.RTASC(JL)) GO TO 185
C
C     STAR AT BEGINNING OF CATALOG
        JQ = 0
        DO 182 K=1,34
          DO 182 L=1,6
            JQ = JQ+1
            TEM(JQ) = ASTRA(K,L)
  182       CONTINUE
        CALL READEC (STARS,121584,162,J2)
        DO 183 K=1,27
          KQ = 28-K
          DO 183 L=1,6
            ASTRA(KQ+7,L) = ASTRA(KQ,L) - TWPI
  183       CONTINUE
        JQ = 0
        DO 184 K=35,68
          DO 184 L=1,6
            JQ = JQ+1
            ASTRA(K,L) = TEM(JQ)
  184       CONTINUE
        JN = 0
        JL = 8
        GO TO 201
  185   IF(RAHI.LE.RTASC(68)) GO TO 201
C
C     STAR NOT IN CURRENT SECTION OF CATALOG
        JN1 = JN+1
        DO 186 KL=JN1,298
          IF(RAHI.LE.RTSV(KL)) GO TO 198
  186     CONTINUE
        IF(RAHI.GT.RTSV(299)) GO TO 192
C
C     STAR IN LAST SECTION OF CATALOG
        CALL READEC (STARS,121176,408,J2)
        JQ = 0
        DO 188 K=35,68
          DO 188 L=1,6
            JQ = JQ+1
            TEM(JQ) = ASTRA(K,L)
  188       CONTINUE
        CALL READEC (STARS,121584,162,J2)
        DO 190 K=1,27
          DO 190 L=1,6
            ASTRA(K+41,L) = ASTRA(K,L)
  190       CONTINUE
        JQ = 0
        DO 191 K=8,41
          DO 191 L=1,6
            JQ = JQ+1
            ASTRA(K,L) = TEM(JQ)
  191       CONTINUE
        JN = 298
        JL = 8
        GO TO 201
C
```

```
C     STAR AT END OF CATALOG
  192   CALL READEC (STARS,121584,162,J2)
        JQ □ 0
        DO 194 K=1,27
          DO 194 L=1,6
            JQ □ JQ+1
            TEM(JQ) □ ASTRA(K,L)
  194       CONTINUE
        CALL READEC (STARS,0,408,J2)
        DO 195 K=1,34
          DO 195 L=1,6
            ASTRA(K+34,L) □ ASTRA(K,L) + TWPI
  195       CONTINUE
        JQ = 0
        DO 197 K=8,34
          DO 197 L=1,6
            JQ = JQ+1
            ASTRA(K,L) = TEM(JQ)
  197       CONTINUE
        JN = 299
        JL = 8
        GO TO 201
C
C     STAR IN GENERAL SECTION OF CATALOG
  198   IF(KL.EQ.1) GO TO 200
        IF(RALO.GT.RTSV(KL-1)) GO TO 200
C
C     STAR RANGE OVERLAPS TWO SECTIONS
        KL1 □ KL-1
        IF(KL1.EQ.JN) GO TO 1983
        IA = (KL1-1)*408
        CALL READEC (STARS,IA,408,J2)
 1983   JQ □ 0
        DO 1985 K=35,68
          DO 1985 L=1,6
            JQ = JQ+1
            TEM(JQ) □ ASTRA(K,L)
 1985       CONTINUE
        IA = (KL-1)*408
        CALL READEC (STARS,IA,408,J2)
        DO 1988 K=1,34
          DO 1988 L=1,6
            ASTRA(K+34,L) = ASTRA(K,L)
 1988       CONTINUE
        JQ = 0
        DO 199 K=1,34
          DO 199 L=1,6
            JQ = JQ+1
            ASTRA(K,L) = TEM(JQ)
  199       CONTINUE
        JN = KL1
        JL = 1
        GO TO 201
C
C     STAR RANGE CONTAINED IN ONE SECTION
  200   IA = (KL-1)*408
        CALL READEC (STARS,IA,408,J2)
        JN = KL
        JL = 1
  201   J = JL-1
C
C     COMPARE APPROXIMATE STAR WITH STAR POSITION IN SECTION
  202   J = J+1
        IF(RTASC(J).LT.RALO) GO TO 202
  220   IF(RTASC(J).GT.RAHI) GO TO 238
C
C     CHECK IF CATALOG STAR IS A POSSIBLE IDENTITY FOR APPROXIMATE STAR
        CALL CHECK (RTASC(J),PMRA(J),DECL(J),PMDC(J),CATNO(J),SRCBK(J),
     1    I,R(18))
        J = J+1
        IF(J.LE.68) GO TO 220
        PRINT 95,COORD(I,5)
        LNCNT = LNCNT + 2
        COORD(I,3) = BLANK
        L = AMOD(COORD(I,5),10.)
        IF(L.EQ.0) L = 10
        NID(L) = NID(L) + 1
        COORD(I,5) □ -COORD(I,5)
        IF(COORD(I,5).EQ.0.) COORD(I,5) □ -1.
        GO TO 239
C
C     CHECK IF CLOSEST STAR IN CATALOG IS A BINARY
C     IF NOT, REPLACE APPROXIMATE STAR COORDINATES IN STORAGE WITH
C     CLOSEST STAR FROM CATALOG
  238   CALL TEST (I)
  239   IF(I.EQ.N) GO TO 240
        IF(LNCNT+3.LE.60) GO TO 240
        PRINT 34,EVENT,IPLT
        LNCNT □ 3
  240   CONTINUE
C
C     PRINT STATISTICS OF STAR IDENTIFICATION
  252 IIT = NIT = 0
      PRINT 49,EVENT,IPLT
      DO 2508 K=1,10
        L = K
        IF(K.EQ.10) L □ 0
        PRINT 50,L,IID(K),NID(K)
        IIT = IIT + IID(K)
        NIT = NIT + NID(K)
 2508   CONTINUE
      PRINT 51,IIT,NIT
      ISGG □ 0
      DO 2511 K=1,JJ
        IF(ISG(K).NE.0) GO TO 2511
        IF(ISGG.NE.0) GO TO 2510
        PRINT 44
        ISGG = 1
 2510   ITRL = XTRL(K)
        PRINT 45,ITRL
 2511   CONTINUE
C
C     SORT STARS ON CATALOG NUMBER
      LEN = 0 $ IFIRST = NEXT = 1
      CALL OBUF (1)
      ENCODE (80,9,BUFFIN(INBUF)) ISTEP,(HEAD(K),K=1,6)
      NT = N
      PRINT 35,EVENT,IPLT
      LNCNT = 3
 2529 NK = NT
      I = 0
 2530 I = I+1
 2531 IF(COORD(I,3).NE.BLANK) GO TO 2533
      DO 2532 L=1,9
        TEMP = COORD(I,L)
        COORD(I,L) = COORD(NK,L)
        COORD(NK,L) = TEMP
 2532   CONTINUE
      NK = NK - 1
      IF(NK.EQ.I) GO TO 255
      GO TO 2531
 2533 J = I
      I1 = I+1
      DO 254 K=I1,NK
        IF(COORD(I,3).NE.COORD(K,3)) GO TO 254
        J = J+1
        IF(J.EQ.K) GO TO 254
        DO 253 L=1,9
            TEMP = COORD(J,L)
            COORD(J,L) = COORD(K,L)
            COORD(K,L) = TEMP
  253       CONTINUE
  254   CONTINUE
      IF(I.LT.(NK-1)) GO TO 2530
  255 NNN = NT
      NCS = 1
      I = 0
C
C     CHECK IF SAME STAR FROM TAPE USED MORE THAN ONCE
  256 I = I+1
  257 NS = 0
      TEMA = COORD(I,1)
      TEMD = COORD(I,2)
      I1 = I+1
  258 IF(COORD(I,5).LT.0.) GO TO 270
      IF(I1.GT.NT) GO TO 260
      IF(COORD(I1,3).NE.COORD(I,3)) GO TO 260
      NS = NS + 1
      TEMA = TEMA + COORD(I1,1)
      TEMD = TEMD + COORD(I1,2)
      NNN = NNN - 1
      I1 = I1 + 1
      GO TO 258
  260 IF(NS.GT.0) GO TO 265
C
C     OUTPUT STAR USED ONLY ONCE
      PRINT 31,COORD(I,1),COORD(I,2),(COORD(I,M),M=7,9),COORD(I,5),
     1  COORD(I,4),COORD(I,3)
      LNCNT = LNCNT + 1
      CALL LINE (LNCNT)
      CALL OBUF (1)
      ENCODE (80,41,BUFFIN(INBUF)) COORD(I,1),COORD(I,2),COORD(I,7),
     1  COORD(I,8),COORD(I,9),COORD(I,5),COORD(I,4),COORD(I,3)
      GO TO 275
C
C     OUTPUT STAR USED MORE THAN ONCE
  265 OBS ◘ 0.
      I2 = I + NS
      DO 268 M=I,I2
        OBS ◘ OBS + COORD(M,4)
  268   CONTINUE
      TEMA = TEMA/FLOAT(NS+1)
      TEMD = TEMD/FLOAT(NS+1)
      FNCS = NCS
      PRINT 31,TEMA,TEMD,(COORD(I,M),M=7,9),FNCS,OBS,COORD(I,3)
      LNCNT = LNCNT + 1
      CALL LINE (LNCNT)
      CALL OBUF (1)
      ENCODE (80,41,BUFFIN(INBUF)) TEMA,TEMD,COORD(I,7),COORD(I,8),
     1  COORD(I,9),FNCS,OBS,COORD(I,3)
      GO TO 275
C
C     OUTPUT UNIDENTIFIED STAR
  270 PRINT 32,COORD(I,5),COORD(I,4)
      LNCNT = LNCNT + 1
      CALL LINE (LNCNT)
      CALL OBUF (1)
      ENCODE (80,42,BUFFIN(INBUF)) COORD(I,5),COORD(I,4)
C
C     LOCATE ASSOCIATED PLATE DATA
  275 DO 2751 M=1,4
        IBD(M) ◘ 0
 2751   CONTINUE
      BP ◘ 0.
      COORD6 = COORD(I,6)
      TRLC = AINT(COORD6/10000.)
      MB ◘ 0
      IF(TRLC.EQ.0.) GO TO 278
      MB = 1
      TEMP = AINT(TRLC*.1)
      IBD(1) ◘ TRLC - TEMP*10.
      TRLC ◘ TEMP
      IF(TRLC.EQ.0.) GO TO 276
      MB = 2
      TEMP = AINT(TRLC*.1)
      IBD(2) = TRLC - TEMP*10.
      TRLC = TEMP
      IF(TRLC.EQ.0.) GO TO 276
      MB ◘ 3
      TEMP = AINT(TRLC*.1)
      IBD(3) ◘ TRLC - TEMP*10.
      TRLC = TEMP
      IF(TRLC.EQ.0.) GO TO 276
      MB = 4
      TEMP = AINT(TRLC*.1)
      IBD(4) = TRLC - TEMP*10.
  276 BP ◘ MB
      DO 277 M=1,MB
```

```
        BD = IBD(M)
        COORD6 □ COORD6 - BD*10.**(M+3)
  277   CONTINUE
  278 K □ COORD6
      K1 = COORD6 - 1. + COORD(I,4) + BP
      IF(I.EQ.NT) GO TO 289
      IF(NS.GT.0) GO TO 282
C
C     OUTPUT PLATE DATA FOR STAR USED ONLY ONCE
      DO 281 M=K,K1
        IF(MB.EQ.0) GO TO 2791
        MJ = M - K + 1
        DO 279 MN = 1,MB
          IF(MJ.EQ.IBD(MN)) GO TO 281
  279     CONTINUE
 2791   PRINT 33,(DATA(M,MN),MN=1,6),COORD(I,5),DATA(M,7)
        LNCNT □ LNCNT + 1
        CALL LINE (LNCNT)
        CALL OBUF(1)
        ENCODE (80,43,BUFFIN(INBUF)) (DATA(M,MN),MN=1,6),COORD(I,5),
     1    DATA(M,7)
  281   CONTINUE
      GO TO 256
C
C     OUTPUT PLATE DATA FOR STAR USED MORE THAN ONCE
  282 DO 285 M=K,K1
        FNCS □ NCS
        IF(MB.EQ.0) GO TO 2831
        MJ = M - K + 1
        DO 283 MN=1,MB
          IF(MJ.EQ.IBD(MN)) GO TO 285
  283     CONTINUE
 2831   PRINT 33,(DATA(M,MN),MN=1,6),FNCS,DATA(M,7)
        LNCNT = LNCNT + 1
        CALL LINE (LNCNT)
        CALL OBUF(1)
        ENCODE (80,43,BUFFIN(INBUF)) (DATA(M,MN),MN=1,6),FNCS,DATA(M,7)
  285   CONTINUE
      IF(I.EQ.I2) GO TO 287
      I = I+1
      GO TO 275
  287 NCS = NCS + 1
      GO TO 256
C
C     OUTPUT LAST SET OF PLATE DATA
  289 FNCC = COORD(I,5)
      IF(NS.GT.0) FNCC = NCS
      IF(MB.EQ.0) GO TO 291
      K1 = K1 + 1   $   MJ = K1 - K + 1
  290 K1 □ K1 - 1   $   MJ □ MJ - 1
      DO 2902 MN=1,MB
        IF(MJ.EQ.IBD(MN)) GO TO 290
 2902   CONTINUE
  291 IF(K.EQ.K1) GO TO 295
      K1 = K1 - 1
      DO 294 M=K,K1
        IF(MB.EQ.0) GO TO 2921
        MJ = M - K + 1
        DO 292 MN=1,MB
          IF(MJ.EQ.IBD(MN)) GO TO 294
  292     CONTINUE
 2921   PRINT 33,(DATA(M,MN),MN=1,6),FNCC,DATA(M,7)
        LNCNT = LNCNT + 1
        CALL LINE (LNCNT)
        CALL OBUF(1)
        ENCODE (80,43,BUFFIN(INBUF)) (DATA(M,MN),MN=1,6),FNCC,DATA(M,7)
  294   CONTINUE
      K1 = K1 + 1
  295 PRINT 33,(DATA(K1,MN),MN=1,6),FNCC,DATA(K1,7),IT
      CALL OBUF (1)
      ENCODE (80,43,BUFFIN(INBUF)) (DATA(K1,MN),MN=1,6),FNCC,
     1  DATA(K1,7),IT
      PRINT 38,NT,NNN
      CALL OBUF(0)
  300 END FILE 5
      GO TO 100
    2 FORMAT(2(1X,E14.8),8X,A5,26X,I1)
    3 FORMAT(3(1X,E14.8))
    4 FORMAT(63X,I3,A5,3I2,I1)
    5 FORMAT(18X,A3,5A10,A9)
    6 FORMAT(1H1,30(/),60X,* JOB STEP *, A3////50X,5A10,A9)
    7 FORMAT(2E14.8,3E10.4,I2,2I3,A5,8X,I1)
    9 FORMAT(18HB        JOB STEP ,A3,5A10,A9)
   10 FORMAT(79X,I1)
   12 FORMAT(3E10.3)
   22 FORMAT(1X,2(1X,E14.7),8X,A5,26X,I1)
   23 FORMAT(1X,3(1X,E14.7))
   24 FORMAT(*T*,63X,I3,A5,3I2,I1)
   25 FORMAT(*T*,3(1X,E14.7))
   26 FORMAT(1X,3E10.3,20X,* NOT FK4 */1X,3E10.3,20X,* FK4 *)
   30 FORMAT(1H1/43H STAR IDENTIFICATION AND UPDATING    EVENT ,A5,
     1  9H   PLATE ,A5,7H   DATE,3I3///7H  INPUT/)
   31 FORMAT(*T*,2E14.7,3E10.3,2X,F6.0,2X,F3.0,1X,A6)
   32 FORMAT(*T*,60X,F6.0,2X,F3.0)
   33 FORMAT(*T*,2F12.10,3E10.3,2F6.0,F11.8,2X,I1)
   34 FORMAT(1H1/14X,41H CATALOG INFORMATION FOR IDENTIFIED STARS,7X,
     1  7H EVENT A5,9H   PLATE A5//)
   35 FORMAT(3HTPG/*T*,12H OUTPUT DATA,11X,6HEVENT A5,9H   PLATE A5/*T*)
   38 FORMAT(/10X,* NUMBER OF STARS BEFORE GROUPING  *I3/10X,33H NUMBER
     1OF STARS AFTER GROUPING   I3)
   41 FORMAT(2E14.7,3E10.3,2X,F6.0,2X,F3.0,1X,A6)
   42 FORMAT(60X,F6.0,2X,F3.0)
   43 FORMAT(2F12.10,3E10.3,2F6.0,F11.8,2X,I1)
   44 FORMAT(*T*/*T*/*T*/*T*,35H THE FOLLOWING TRAILS WERE NOT USED)
   45 FORMAT(*T*,5X,I3)
   46 FORMAT(///49H   HEADERS AND DATA CHECKED FOR SAME PLATE NUMBER//)
   47 FORMAT(1H1,37X,36H MEANED APPROXIMATE STAR COORDINATES,20X,7H EVEN
     1T A5,9H   PLATE A5/)
   48 FORMAT(1X,3(2X,F11.8,1X,F11.8,1X,F4.0,F2.0,7X))
   49 FORMAT(1H1/51X,7H EVENT A5,9H   PLATE A5/8X,5H STAR,11X,6H STARS,
     1  9X,6H STARS/4X,13H TRAIL NUMBER,4X,11H IDENTIFIED,3X,
     2  13H UNIDENTIFIED/)
   50 FORMAT(10X,I1,14X,I3,12X,I3)
   51 FORMAT(/16X,7H TOTALS,I5,12X,I3)
```

```
87 FORMAT(/5X,14H STAR NUMBER  F4.0,3X,*  HAS APPROXIMATE RIGHT ASCEN
  1SION GREATER THAN 24 HOURS *)
90 FORMAT(/20H MORE THAN 90 TRAILS)
91 FORMAT(//42H HEADERS AND DATA ARE FOR DIFFERENT PLATES//
  1  11H   HEADERS A5,8X,6H DATA A5)
92 FORMAT(27H NO HEADER CARDS FOR TRAIL F3.0)
93 FORMAT(/22H MORE THAN 1000 POINTS)
94 FORMAT(/20H MORE THAN 200 STARS)
95 FORMAT(* TOLERANCE RANGE FOR STAR *,F4.0,* EXCEEDS 68 STARS ON TAP
  1E*/* IF THIS HAPPENS OFTEN, SEE PROGRAMMER ABOUT CHANGING LIMITS*)
96 FORMAT(1HT)
97 FORMAT(4X,8A10)
98 FORMAT(*1  JOB STEP ABORTED--  INPUT AREA */)
99 FORMAT(3HTPG)
   END
```

```
      SUBROUTINE LINE (LNCNT)
      IF (LNCNT.LT.60) GO TO 257
      PRINT 99
      LNCNT = 1
  257 RETURN
   99 FORMAT (3HTPG)
      END
      SUBROUTINE CHECK (ALPH,PMR,DELT,PMD,CAT,SRC,I,TT)
      COMMON /CHK/ C,DTOL,PR1DIF,PR2DIF,ST,FWT,INDWT
      DIMENSION C(200,9),ST(2,11),FWT(3,2)
      DATA (SRCBK=77777700000000000000B),(FK4=3HF4 ),(W1=1.),(W2=1.),
     1  (W3=0.),(FILL=00000055555555555555B)
C
C     CHECK IF CATALOG STAR DECLINATION IS WITHIN TOLERANCE RANGE
      IF(ABS(DELT-C(I,2)).GT.DTOL) GO TO 10
C
C     UPDATE CATALOG STAR POSITION TO TIME OF OBSERVATION
      RT = ALPH
      PA = PMR*1.E-10
      DEC = DELT
      PD □ PMD*1.E-10
      CALL UPDATE (RT,PA,DEC,PD,C(I,5),TT)
C
C     CHECK CATALOG STAR AGAINST PREVIOUSLY CHOSEN CATALOG STARS, IF ANY
      DDIF = ABS(DEC - C(I,3))
      IF(DDIF.GE.PR2DIF) GO TO 10
      IF(INDWT.NE.0) GO TO 1
      ST(2,4) = W1
      ST(2,5) = W2
      ST(2,6) = W3
      GO TO 3
    1 ABC = SRC.AND.SRCBK.OR.FILL
      IF(ABC.EQ.FK4) GO TO 2
      ST(2,4) = FWT(1,1)
      ST(2,5) = FWT(2,1)
      ST(2,6) = FWT(3,1)
      GO TO 3
    2 ST(2,4) = FWT(1,2)
      ST(2,5) = FWT(2,2)
      ST(2,6) = FWT(3,2)
    3 ST(2,1) □ RT
      ST(2,2) = DEC
      ST(2,3) = CAT
      ST(2,7) = SRC
      ST(2,8) □ ALPH
      ST(2,9) □ PA
      ST(2,10) = DELT
      ST(2,11) □ PD
      PR2DIF = DDIF
      IF(DDIF.GE.PR1DIF) GO TO 10
      DO 5 K=1,11
        TEMP = ST(2,K)
        ST(2,K) = ST(1,K)
        ST(1,K) = TEMP
    5   CONTINUE
      TEMP = PR2DIF
      PR2DIF = PR1DIF
      PR1DIF □ TEMP
   10 RETURN
      END
      SUBROUTINE UPDATE (RA,PMRA,DEC,PMDC,CST,TT)
      COMMON /UPD/ DTDIF,IDIFF,IDCR,JJ,XTRL,DN,NARC,QJAY
      COMMON /TST/ IDUM(20),LNCNT
      DIMENSION XJAY(21,25),QJAY(525),XTRL(90),DN(90,8),Q(3),DC(3,3),
     1  QQ(3),ST(2,11)
      EQUIVALENCE (XJAY,QJAY)
      DATA (RSEC=.4848136811O953E-5),(RTSEC=.7272205216643E-4),
     1  (RDG=.17453292519943E-1)
      IF(DEC)134,133,133
  133 IDEC = 1
      GO TO 135
  134 IDEC = -1
C
C     UPDATE TO BEGINNING OF YEAR OF OBSERVATION
  135 X = COS(RA)*COS(DEC)
      Y = SIN(RA)*COS(DEC)
      Z = SIN(DEC)
      PMX = -PMRA*COS(DEC)*SIN(RA) - PMDC*COS(RA)*SIN(DEC)
      PMY =  PMRA*COS(DEC)*COS(RA) - PMDC*SIN(RA)*SIN(DEC)
      PMZ =  PMDC*COS(DEC)
      PMSQ = PMRA*PMRA*COS(DEC)*COS(DEC) + PMDC*PMDC
      PMXD = -X * PMSQ
      PMYD = -Y * PMSQ
      PMZD = -Z * PMSQ
      TSQ = .5*TT**2
      Q(1) = X + PMX*TT + PMXD*TSG
      Q(2) = Y + PMY*TT + PMYD*TSQ
      Q(3) = Z + PMZ*TT + PMZD*TSQ
      T = TT/100.
      TSQ = T • T
      TCU = TSQ * T
      DELTA = 2304.948*T + .302*TSQ + .018*TCU
      ZETA = (DELTA + .791*TSQ) * RSEC
      DELTA = DELTA * RSEC
      THETA = (2004.2555*T - .426*TSQ - .042*TCU) * RSEC
      DC(1,1) =  COS(DELTA)*COS(THETA)*COS(ZETA) - SIN(DELTA)*SIN(ZETA)
      DC(1,2) = -SIN(DELTA)*COS(THETA)*COS(ZETA) - COS(DELTA)*SIN(ZETA)
      DC(1,3) = -SIN(THETA)*COS(ZETA)
      DC(2,1) =  COS(DELTA)*COS(THETA)*SIN(ZETA) + SIN(DELTA)*COS(ZETA)
      DC(2,2) = -SIN(DELTA)*COS(THETA)*SIN(ZETA) + COS(DELTA)*COS(ZETA)
      DC(2,3) = -SIN(THETA)*SIN(ZETA)
      DC(3,1) =  COS(DELTA)*SIN(THETA)
      DC(3,2) = -SIN(DELTA)*SIN(THETA)
      DC(3,3) =  COS(THETA)
      DO 136 I=1,3
        QQ(!) = 0.
        DO 136 K=1,3
          QQ(I) = QQ(I) + DC(I,K)*Q(K)
  136   CONTINUE
      CALL ANGLE(QQ(2),QQ(1),RTASC)
      SQ = SQRT(QQ(1)*QQ(1) + QQ(2)*QQ(2))
      DECL = ATAN(QQ(3)/SQ)
      IF(ABS(QQ(3)/SQ).LT.1.E+4) GO TO 138
      PRINT 1,CST
      LNCNT = LNCNT + 4
      RETURN
C
```

```
C     LOCATE PARAMETERS (INDEPENDENT DAY NUMBERS) FOR INITIAL OBSERVA-
C     TION OF THE STAR
  138 JTRL = AMOD(CST,10.)
      DO 139 I=1,JJ
        KTRL = XTRL(I)/100.
        IF(KTRL.EQ.JTRL) GO TO 140
  139   CONTINUE
      PRINT 2,JTRL
      LNCNT □ LNCNT + 3
      RETURN
C
C     COMPUTE J-CORRECTIONS
  140 RA □ RTASC/RDG/15.
      IF(RA.GE.12.) RA = RA - 12.
      KRA = RA
      RADIF □ RA - FLOAT(KRA)
      KRA = KRA + 2
      KRA1 = KRA + 1
      IF(KRA.EQ.13) KRA1 □ 2
      CDIF = DTDIF * RADIF
      IF(IDEC-IDCR)143,146,144
  143 NARC = NARC + 26
      GO TO 145
  144 NARC = NARC - 26
  145 IA □ (NARC-1)*525 + 121750
      CALL READEC (QJAY,IA,525,J2)
  146 IDCR = IDEC
      CORRJ □ XJAY(IDIFF,KRA)*(1.-DTDIF-RADIF+CDIF) + XJAY(IDIFF+1,KRA)
     1 *(DTDIF-CDIF) + XJAY(IDIFF,KRA1)*(RADIF-CDIF) + XJAY(IDIFF+1,
     2 KRA1)*CDIF
      CORRJP = XJAY(IDIFF,KRA+12)*(1.-DTDIF-RADIF+CDIF) + XJAY(IDIFF+1,
     1 KRA+12)*(DTDIF-CDIF) + XJAY(IDIFF,KRA1+12)*(RADIF-CDIF) + XJAY
     2 (IDIFF+1,KRA1+12)*CDIF
      CORRJ = CORRJ*1.E-5*RTSEC
      CORRJP □ CORRJP*1.E-4*RSEC
C
C     UPDATE TO TIME OF OBSERVATION
      RA □ RTASC + DN(I,4) + DN(I,8)*PMRA + DN(I,5)*TAN(DECL)*SIN(RTASC
     1 + DN(I,2)) + DN(I,6)*SIN(RTASC+DN(I,3))/COS(DECL) + CORRJ *
     2 TAN(DECL)*TAN(DECL)
      DEC □ DECL + DN(I,8)*PMDC + DN(I,6)*SIN(DECL)*COS(RTASC+DN(I,3)) +
     1 DN(I,5)*COS(RTASC+DN(I,2)) + DN(I,7)*COS(DECL) + CORRJP *
     2 TAN(DECL)
      RETURN
    1 FORMAT(/13H STAR NUMBER ,F4.0,28H HAS DECLINATION = 90 OR -90/
     1 22H  STAR CANNOT BE USED /)
    2 FORMAT(/27H NO HEADER CARDS FOR TRAIL I1/)
      END
      SUBROUTINE TEST (I)
      COMMON /CHK/ C,DTOL,PR1DIF,PR2DIF,ST,FWT(3,2),INDWT
      COMMON /TST/ NID,IID,LNCNT
      DIMENSION ST(2,11),C(200,9),NID(10),IID(10)
      DATA(TWSC=.000096962736),(BLANK=5H     )
      KO = C(I,4)
      KR = C(I,5)
C
C     STAR UNIDENTIFIED OR BINARY
    1 IF(PR1DIF.LT.7.) GO TO 2
      PRINT 11,KR
      LNCNT □ LNCNT + 3
      GO TO 6
    2 IF(PR2DIF.GT.7.) GO TO 4
      DLTDEC □ ABS(ST(1,2) - ST(2,2))
      IF(DLTDEC.GT.TWSC) GO TO 4
      PRINT 12,KR
      LNCNT = LNCNT + 3
    6 C(I,3) = BLANK
      J = AMOD(C(I,5),10.)
      IF(J.EQ.0) J = 10
      NID(J) = NID(J) + 1
      C(I,5) = -C(I,5)
      IF(C(I,5).EQ.0.) C(I,5) □ -1.
      GO TO 8
C
C     NORMAL IDENTIFICATION
    4 C(I,1) = ST(1,1)
      C(I,2) = ST(1,2)
      C(I,3) □ ST(1,3)
      C(I,7) = ST(1,4)
      C(I,8) = ST(1,5)
      C(I,9) = ST(1,6)
      PRINT 13,(ST(1,K),K=8,11),ST(1,3),KR,KO,ST(1,7)
      LNCNT = LNCNT + 1
      J = AMOD(C(I,5),10.)
      IF(J.EQ.0) J = 10
      IID(J) = IID(J) + 1
    8 RETURN
   11 FORMAT(/48X,14H STAR NUMBER  I4,3X,15H NOT IDENTIFIED/)
   12 FORMAT(/48X,14H STAR NUMBER  I4,3X,10H IS BINARY/)
   13 FORMAT(1X,2(1X,F11.8,1X,F13.10),2X,A5,2X,I4,I2,2X,A9)
   20 FORMAT(1H1/14X,41H CATALOG INFORMATION FOR IDENTIFIED STARS,7X,
     1  7H EVENT A5,9H   PLATE A5//)
      END
   SUBROUTINE ANGLE (Y,X,A)
   PI = 3.1415926535898
   IF(X) 30,20,10
10 IF(Y.LT.0.) GO TO 13
   A = ATAN(Y/X)
   GO TO 40
13 A = 2.*PI + ATAN(Y/X)
   GO TO 40
20 IF (Y) 23,22,21
21 A □ PI/2.
   GO TO 40
22 A = 0.
   GO TO 40
23 A = 3.*PI/2.
   GO TO 40
30 A = PI + ATAN(Y/X)
40 RETURN
   END
```

```
      SUBROUTINE DBUF (NORECS)
C
      COMMON /CDBUF/ LENGT ,NEXT,IFIRST,IXBUF,BUFF(1024),ENDFLQ,KS,
     1  ITI,ITO
      DATA (ICOUNT=0)
      DATA (IEF=1000000000000B),(IPR=2000000000000000B)
C
      IF (NORECS) 70,10,70
   10 CALL LTRIO (ITI,111B,BUFF(IFIRST),BUFF(IFIRST+511),KS)
      IF (KS .LT. 0) GO TO 76
      IF ((KS.AND.IRR) .NE. 0) PRINT 100
      IF ((KS.AND.IEF) .NE. 0) GO TO 60
      IFIRST = MOD(IFIRST+512,1024)
      ICOUNT = 0
      RETURN
   60 ICOUNT = ICOUNT+1
      IF (ICOUNT .LT. 2) GO TO 10
      PRINT 64
      CALL LTRIO (ITO,115B,A,B,JS)
      STOP
   70 IF (MOD(LENGT,512)) 79,74,79
   74 CALL LTRIO (ITI,111B,BUFF(IFIRST),BUFF(IFIRST+511),KS)
      IFIRST = MOD(IFIRST+512,1024)
   79 LENGT  = MOD(LENGT+8*NORECS,1024)
      IXBUF = NEXT
      NEXT = LENGT +1
      RETURN
   76 PRINT 77,KS
      STOP
  100 FORMAT(/ * TROUBLE IN INPUT TAPE • /)
   64 FORMAT (*1  JOB TERMINATED--  END OF DATA*)
   77 FORMAT (*1  JOB ABORTED-- STATUS WORD   * ,020)
      END
      SUBROUTINE OBUF (NORECS)
      COMMON /CDBUF/ LENGT,NEXT,IFIRST,IXBUF,BUFF(1024),ENDFLQ,JSAP,
     1  ITI,ITO
C
      IF (NORECS) 10,40,10
   10 LENGT = LENGT + NORECS
      IF(LENGT.GT.64) GO TO 20
      IXBUF = NEXT
      GO TO 30
   20 CALL LTRIO (ITO,112B,BUFF(IFIRST),BUFF(IFIRST+511),KS)
      IF (KS .LT. 0) GO TO 80
      IXBUF = IFIRST □ MOD(IFIRST+512,1024)
      LENGT □ NORECS
   30 NEXT □ IXBUF + NORECS*8
      RETURN
   40 INDEX = IFIRST+(LENGT-NORECS)*8-1
      CALL LTRIO (ITO,112B,BUFF(IFIRST),BUFF(INDEX),KS)
      IF (KS .LT. 0) GO TO 80
      ENDFLQ □ 0.
      RETURN
   80 PRINT 100,KS
      STOP
  100 FORMAT (*1  JOB ABORTED-- STATUS WORD   * ,O20)
      END
```

U.S. GOVERNMENT PRINTING OFFICE: 1979—O-507-908

(Continued from inside front cover)

NOAA TECHNICAL REPORTS

NOS 41 A User's Guide to a Computer Program for Harmonic Analysis of Data at Tidal Frequencies. R. E. Dennis and E. E. Long, July 1971. Price $0.65 (COM-71-50606)

NOS 42 Computational Procedures for the Determination of a Simple Layer Model of the Geopotential From Doppler Observations. Bertold U. Witte, April 1971. Price $0.65 (COM-71-50400)

NOS 43 Phase Correction for Sun-Reflecting Spherical Satellite. Erwin Schmid, August 1971. Price $0.25 (COM-72-50080)

NOS 44 The Determination of Focal Mechanisms Using P- and S-Wave Data. William H. Dillinger, Allen J. Pope, and Samuel T. Harding, July 1971. Price $0.60 (COM-71-50392)

NOS 45 Pacific SEAMAP 1961-70 Data for Area 15524-10: Longitude 155°W to 165°W, Latitude 24°N to 30°N, Bathymetry, Magnetics, and Gravity. J. J. Dowling, E. E. Chiburis, P. Dehlinger, and M. J. Yellin, January 1972. Price $3.50 (COM-72-51029)

NOS 46 Pacific SEAMAP 1961-70 Data for Area 15530-10: Longitude 155°W to 165°W, Latitude 30°N to 36°N, Bathymetry, Magnetics, and Gravity. J. J. Dowling, E. F. Chiburis, P. Dehlinger, and M. J. Yellin, January 1972. Price $3.50

NOS 47 Pacific SEAMAP 1961-70 Data for Area 15248-14: Longitude 152°W to 166°W, Latitude 48°N to 54°N, Bathymetry, Magnetics, and Gravity. J. J. Dowling, E. F. Chiburis, P. Dehlinger, and M. J. Yellin, April 1972. Price $3.50 (COM-72-51030)

NOS 48 Pacific SEAMAP 1961-70 Data for Area 16648-14: Longitude 166°W to 180°, Latitude 48°N to 54°N, Bathymetry, Magnetics, and Gravity. J. J. Dowling, E. F. Chiburis, P. Dehlinger, and M. J. Yellin, April 1972. Price $3.00 (COM-72-51028)

NOS 49 Pacific SEAMAP 1961-70 Data for Areas 16530-10 and 17530-10: Longitude 165°W to 180°, Latitude 30°N to 36°N, Bathymetry, Magnetics, and Gravity. E. F. Chiburis, J. J. Dowling, P. Dehlinger, and M. J. Yellin, July 1972. Price $4.75

NOS 50 Pacific SEAMAP 1961-70 Data for Areas 16524-10 and 17524-10: Longitude 165°W to 180°, Latitude 24°N to 30°N, Bathymetry, Magnetics, and Gravity. E. F. Chiburis, J. J. Dowling, P. Dehlinger, and M. J. Yellin, July 1972. Price $5.75

NOS 51 Pacific SEAMAP 1961-70 Data for Areas 15636-12, 15642-12, 16836-12, and 16842-12: Longitude 156°W to 180°, Latitude 36°N to 48°N, Bathymetry, Magnetics, and Gravity. E. F. Chiburis, J. J. Dowling, P. Dehlinger, and M. J. Yellin, July 1972. Price $11.00 (COM-73-50280)

NOS 52 Pacific SEAMAP 1961-70 Data Evaluation Summary. P. Dehlinger, E. F. Chiburis, and J. J. Dowling, July 1972. Price $0.40

NOS 53 Grid Calibration by Coordinate Transfer. Lawrence W. Fritz, November 1972. (COM-73-50240)

NOS 54 A Cross-Coupling Computer for the Oceanographer's Askania Gravity Meter. Carl A. Pearson and Thomas E. Brown, November 1972. (COM-73-50317)

NOS 55 A Mathematical Model for the Simulation of a Photogrammetric Camera Using Stellar Control. Chester C Slama, December 1972.

NOS 56 Cholesky Factorization and Matrix Inversion. Erwin Schmid, March 1973. (COM-73-50486)

NOS 57 Complete Comparative Calibration. Lawrence W. Fritz, in press, 1973.

NOS 58 Telemetering Hydrographic Tide Gauge. Charles W. Iseley, in press, 1973.

NOS 59 Gravity Gradients at Satellite Altitudes. B. Chovitz, J. Lucas, and F. Morrison, in press, 1973.